Renata Vieira Paulo Quaresma
Maria das Graças Volpe Nunes
Nuno J. Mamede Cláudia Oliveira
Maria Carmelita Dias (Eds.)

LNAI 3960

Computational Processing of the Portuguese Language

7th International Workshop, PROPOR 2006
Itatiaia, Brazil, May 2006
Proceedings

Springer

Lecture Notes in Artificial Intelligence 3960

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Renata Vieira Paulo Quaresma
Maria das Graças Volpe Nunes
Nuno J. Mamede Cláudia Oliveira
Maria Carmelita Dias (Eds.)

Computational Processing of the Portuguese Language

7th International Workshop, PROPOR 2006
Itatiaia, Brazil, May 13-17, 2006
Proceedings

Springer

Volume Editors

Renata Vieira
Universidade do Vale de Rio de Sinos, UNISINOS, São Leopoldo, Brazil
E-mail: renata@unisinos.br

Paulo Quaresma
Universidade de Évora, Portugal
E-mail: pq@di.uevora.pt

Maria das Graças Volpe Nunes
Universidade de São Paulo, USP, São Carlos, SP, Brazil
E-mail: gracan@icmc.usp.br

Nuno J. Mamede
Universidade Técnica de Lisboa, L2F, INESC-ID, Lisboa, Portugal
E-mail: Nuno.Mamede@inesc-id.pt

Cláudia Oliveira
Instituto Militar de Engenharia, Rio de Janeiro, Brazil
E-mail: cmaria@de9.ime.eb.br

Maria Carmelita Dias
Pontifícia Universidade Católica do Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil
E-mail: mcdias@let.puc-rio.br

Library of Congress Control Number: 2006925088

CR Subject Classification (1998): I.2.7, F.4.2-3, I.2, H.3, I.7

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-34045-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-34045-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11751984 06/3142 5 4 3 2 1 0

Preface

Since 1993, PROPOR Workshops have become an important forum for researchers involved in the Computational Processing of Portuguese, both written and spoken. This PROPOR Workshop follows previous workshops held in 1993 (Lisbon, Portugal), 1996 (Curitiba, Brazil), 1998 (Porto Alegre, Brazil), 1999 (Évora, Portugal), 2000 (Atibaia, Brazil) and 2003 (Faro, Portugal). The workshop has increasingly contributed to bringing together researchers and industry partners from both sides of the Atlantic.

The constitution of an international Program Committee and the adoption of high-standard referee procedures demonstrate the steady development of the field and of its scientific community.

In 2006 PROPOR received 56 paper submissions from 11 different countries: Brazil, Portugal, Spain, Norway, USA, Italy, Japan, France, Canada, Denmark and the UK, from which 9 are represented in the accepted papers.

Each submitted paper underwent a careful, triple-blind review by the Program Committee. All those who contributed are mentioned in the following pages. The reviewing process led to the selection of 20 regular papers for oral presentation and 17 short papers for poster sections, which are published in this volume.

The workshop and this book were structured around the following main topics, seven for full papers: (i) automatic summarization; (ii) resources; (iii) automatic translation; (iv) named entity recognition; (v) tools and frameworks; (vi) systems and models; and another five topics for short papers; (vii) information extraction; (viii) speech processing; (ix) lexicon; (x) morpho-syntactic studies; (xi) web, corpus and evaluation.

We are especially grateful to our invited speakers, Adam Kilgarriff (Visiting Research Fellow at the Department of Informatics, University of Sussex) and Marcelo Finger (IME, University of São Paulo), for their invaluable contribution, which undoubtedly increased the interest in the workshop and its quality.

We are indebted to all members of our Technical Program Committee and additional reviewers, as mentioned in the following pages.

May 2006

Renata Vieira
Paulo Quaresma
Maria das Graças Volpe Nunes
Nuno J. Mamede
Claudia Oliveira
Maria Carmelita Dias

Organization

Conference Chairs

Claudia Oliveira (Instituto Militar de Engenharia, Brazil)
Maria Carmelita Dias (Pontifícia Universidade Católica do Rio de Janeiro, Brazil)

Program Co-chair

Renata Vieira (Universidade do Vale do Rio dos Sinos, Brazil)
Paulo Quaresma (Universidade de Évora, Portugal)

Publication Chairs

Renata Vieira (Universidade do Vale do Rio dos Sinos, Brazil)
Paulo Quaresma (Universidade de Évora, Portugal)
Maria das Graças Volpe Nunes (Universidade de São Paulo, Brazil)
Nuno Mamede (Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento, Portugal)

Program Committee

Aldebaro Klautau (Universidade Federal do Pará, Brazil)
Aline Villavicencio (Universidade Federal do Rio Grande do Sul, Brazil)
Ana Frankenberg-Garcia (Instituto Superior de Línguas e Administração, Portugal)
António Branco (Universidade de Lisboa, Portugal)
António Ribeiro (European Commission, Joint Research Centre, Italy)
António Teixeira (Universidade do Aveiro, Portugal)
Ariadne Carvalho (Universidade de Campinas, Brazil)
Belinda Maia (Universidade do Porto, Portugal)
Bento Carlos Dias da Silva (Universidade Estadual Paulista, Brazil)
Berthold Crysmann (German Research Centre for Artificial Intelligence, Germany)
Carlos Augusto Prolo (Pontifícia Universidade Católica do Rio Grande do Sul, Brazil)
Caroline Hagège (Xerox Research Centre, France)
Claudia Oliveira (Instituto Militar de Engenharia, Brazil)
Diana Santos (The Foundation for Scientific and Industrial Research at the Norwegian Institute of Technology, Norway)
Diamantino Freitas (Universidade do Porto, Portugal)
Eckhard Bick (Southern Denmark University, Denmark)
Egidio Terra (Pontifícia Universidade Católica do Rio Grande do Sul, Brazil)
Elisabete Ranchhod (Universidade de Lisboa, Portugal)
Eric Laporte (University of Marne-la-Vallée, France)

Additional Reviewers

Table of Contents

Summarization

Resources

Translation

Named Entity Recognition

Tools and Frameworks

Systems and Models

Information Extraction

Speech Processing

Lexicon

Morpho-syntactic Studies

Web, Corpus and Evaluation

Modeling and Evaluating Summaries Using Complex Networks

Thiago Alexandre Salgueiro Pardo[1], Lucas Antiqueira[1],
Maria das Graças Volpe Nunes[1], Osvaldo N. Oliveira Jr.[1,2],
and Luciano da Fontoura Costa[2]

[1] Núcleo Interinstitucional de Lingüística Computacional (NILC),
CP 668 – ICMC-USP, 13.560-970 São Carlos, SP, Brasil
http://www.nilc.icmc.usp.br
[2] Instituto de Física de São Carlos,
CP 369 – IFSC-USP, 13.560-970 São Carlos, SP, Brasil
{taspardo,lantiq}@gmail.com, gracan@icmc.usp.br,
{chu,luciano}@if.sc.usp.br
http://www.ifsc.usp.br/

Abstract. This paper presents a summary evaluation method based on a complex network measure. We show how to model summaries as complex networks and establish a possible correlation between summary quality and the measure known as dynamics of the network growth. It is a generic and language independent method that enables easy and fast comparative evaluation of summaries. We evaluate our approach using manually produced summaries and automatic summaries produced by three automatic text summarizers for the Brazilian Portuguese language. The results are in agreement with human intuition and showed to be statistically significant.

1 Introduction

Automatic text summarization is the task of automatically producing a shorter version of a text (Mani, 2001), which should convey the essential meaning of the source text and attend the reader's goals. Nowadays, due to the increasing amount of available information, mainly on-line, and the necessity of retrieving such information with high accuracy and of understanding it faster than ever, automatic summarization is unquestionably an important task.

Summaries are present in a wide range of our daily activities. During scientific papers writing, we have to write abstracts; when reading these papers, abstracts help us to determine whether the paper is important or not for our purposes. In a bookshop, the decision of buying a book is usually based on its cover synthesis. Some internet search engines use summaries to identify documents main parts and to help users in choosing which documents to retrieve.

In spite of the extensive investigation into methods for automatic summarization, it is still hard to determine which method is better. Summary evaluation remains an unresolved issue. Various aspects in summaries require evaluation (Mani, 2001), including amount of information, coherence, cohesion, thematic progression, legibility, grammaticality and textuality. Some are hard to define, while some

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 1 – 10, 2006.

significantly overlap. Depending on the final use of a summary, be it for humans or computer applications, different criteria need to be matched: if humans are the intended readers, coherence and cohesion may be necessary; if the summary is to be used in a computer application, sometimes only the depicted information may be enough. There are several summary evaluation metrics, whose computation may be carried out either by humans or computers: if humans perform the evaluation, it becomes expensive, time consuming and prone to errors and inconsistencies; if computers perform it, subjective aspects of the evaluation are lost and evaluation may not be complete. Given the importance of the task, international conferences have been devoted to the theme, with DUC (Document Understanding Conference) being the most prominent, driving research in this area for the past 7 years.

Concomitantly, recent trends in Natural Language Processing (NLP) show the use of graphs as a powerful technique for modeling and processing texts. Such interest in graphs is due to their generic applicability, often leading to elegant solutions to difficult problems. For text summarization purposes, graphs have been used for both summary production (see, e.g., Erkan and Radev, 2004; Mihalcea, 2005) and evaluation (see, e.g., Santos Jr. et al., 2004). In particular, a special kind of graphs, called complex networks, has received great attention over the last few years. They have been proven useful to model NLP and Linguistics problems, in addition to many other applications (see, e.g., Barabási, 2003). Complex networks have been used, for instance, in modeling lexical resources (Sigman and Cecchi, 2002), human-induced words association (Costa, 2004), language evolution modeling (Dorogovtsev and Mendes, 2002), syntactic relationship between words (Cancho et al., 2005) and text quality measurement (Antiqueira et al., 2005a, 2005b).

This paper presents a first approach to the use of complex networks in summary evaluation. Particularly, it builds on the work of Antiqueira et al. (2005a, 2005b), by describing a possible representation of summaries as complex networks and establishing a correlation between summary quality and one of the network properties, namely the dynamics of the network growth. We evaluate our approach using TeMário corpus (Pardo and Rino, 2003), comprising 100 texts in Brazilian Portuguese and the corresponding human-produced (manual) summaries, and automatic summaries produced by the systems GistSumm (Pardo et al., 2003), SuPor (Módolo, 2003) and GEI (Pardo and Rino, 2004).

In the next section, complex networks are introduced. Section 3 describes how summaries are modeled as complex networks. Experimental results with manual and automatic summaries for verifying the correlation of the dynamics of the network growth property and quality are shown in Section 4. Section 5 presents the conclusions and final remarks.

2 Complex Networks: An Overview

Complex networks are particularly complex types of graphs, i.e. structures that contain nodes and edges connecting them. They have received enormous attention in the last few years, but their study can be traced back to initial development in graph theory. However, in contrast to simple graphs, complex networks present connecting structures that tend to depart from being randomly uniform, i.e., their growth is usually not uniformly random (Barabási, 2003). Complex networks have been used to describe several world phenomena, from social networks to internet topology. Such phenomena present properties that often conform to the complex network characteristics, which caused the complex networks to be studied in a wide range of sciences, mainly by mechanical statistics and physics. See Barabási (2003) and Newman (2003) for a comprehensive scenario of complex network uses.

Some properties that may be observable in complex networks are worth mentioning. Networks known as *small world networks* point to the fact that there is a relatively short path between most nodes in the networks. For instance, social networks are usually small worlds. The *clustering coefficient* indicates the tendency of the network nodes to form groups; in a social network, the friends of a person tend to be friends too. A network is said to be *scale free* if the probability of a node having k edges connecting it to other nodes follows a power law distribution, i.e., $P(k){\sim}k^{-\gamma}$, where γ is a constant value dependent on the network properties (topology and connectivity factors, for instance). Scale free networks contain *hubs*, which consist of highly connected nodes. In internet, for example, hubs are the pages receiving links from many other pages. These properties are also applicable to NLP related tasks. Sigman and Cecchi (2002) modeled WordNet (Miller, 1985) as a complex network, where nodes represent the word meanings and edges represent the semantic relations between them. They showed that this network is a small world and contains hubs, mainly because of polysemic words. Motter et al. (2002) modeled a thesaurus as a network, where nodes represent words and edges represent the synonym relations between them, and detected that this network was scale free. Antiqueira et al. (2005a, 2005b) modeled texts as complex networks, where nodes represent the words and edges connect adjacent words in a text. Among other things, they suggested that text quality is somewhat related to the clustering coefficient, with quality deteriorating with an increasing coefficient.

In the next section, we show how to model summaries as complex networks.

3 Representing Summaries as Complex Networks

Our representation of summaries as complex networks follows the scheme proposed by Antiqueira et al. (2005a, 2005b). Firstly, pre-processing steps are carried out: the summary stopwords are removed and the remaining words are lemmatized. Removing stopwords eliminates irrelevant and very common words; using lemmas instead of words causes the processing to be more intelligent, since it is possible to identify words with related meaning. The pre-processed summary is then represented as a complex network. Each word corresponds to a node in the network and words associations are represented as directed edges. In the representation adopted, each association is determined by a simple adjacency relation: for each pair of adjacent words in the summary there is a directed edge in the network pointing from the node that represents the first word to the node representing the subsequent word in the summary. The edges are weighted with the number of times the adjacent words are found in the summary. Significantly, in this representation, sentence and paragraph boundaries are not taken into consideration. As an example, the sample summary of Figure 1 (in Portuguese) is represented by the network in Figure 2.

Lula e Fernando Henrique Cardoso estão nitidamente à frente nas eleições. Nas sondagens anteriores, o segundo lugar de Fernando Henrique era disputado por mais quatro. Hoje, quem mais o ameaça, mesmo assim sem perigo, é Sarney.

Fig. 1. Sample summary

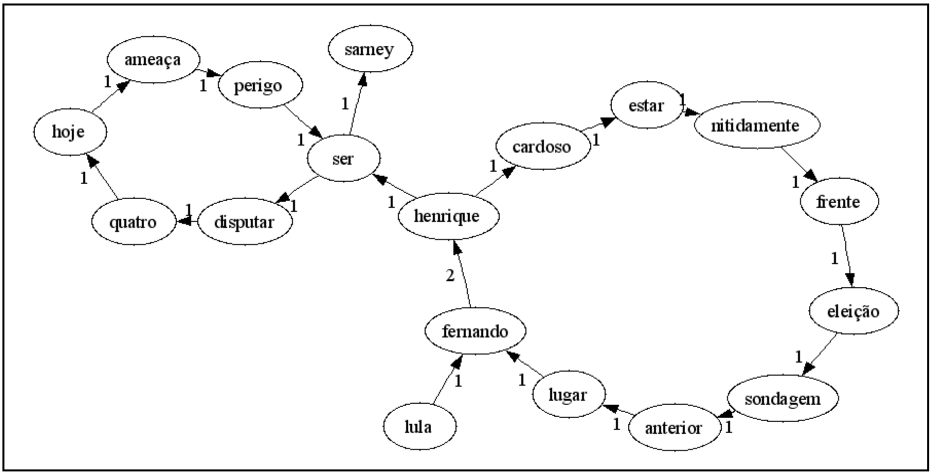


Fig. 2. Complex network for the summary in Figure 1

4 Summary Evaluation

Antiqueira et al. (2005a, 2005b) showed the existence of correlation between the dynamics of network growth and the quality of the text represented. The dynamics of a network growth is a temporal measure of how many connected components there are in the network as words associations are progressively incorporated into the network as it is constructed. Initially, in a time t_0, all N different words (nodes of the network) in the text under analysis are the components. In a subsequent time t_1, when an association is found between any two adjacent words w_i and w_j in the text, there are N-1 components, i.e., the component formed by w_i and w_j and the other N-2 words without any edge between them. This procedure is considered with each new word being added, until only one component representing the whole text is formed. For each text, Antiqueira et al. plot a graphic whose curve indicates the number of components in the network as new words associations are considered (which implies inserting a new edge, if it does not exist, or increasing the edge weight by 1 if it already exists). Considering a straight line in this graphic, which would indicate that there is a linear variation of the number of components as new words associations are considered, the authors showed that good-quality texts tend to be associated to a straight line in the dynamics plot. Moreover, text quality decreased with an increase in the deviation from the straight line.

The general deviation from the straight line for a text is quantified by following formula:

$$deviation = \frac{\sum_{M=1}^{A} \left| f(M) - g(M) \right| / N}{A}$$

where f(M) is the function that determines the number of components for M words associations and g(M) is the function that determines the linear variation of components for M words associations; N is the number of different words in the text and A is the total number of words associations found.

Figure 3 shows the plot for a longer version of the summary in Figure 1, which is a manual summary built by a professional abstractor. The straight dotted line is the one that assume linear variation of the number of components; the other line is the real curve for the summary. According to the above formula, the general deviation for the summary is 0.023. Figure 4 shows the plot for an automatic summary known to be worse, with same size and for the same source text of the summary of Figure 3. Its general deviation is 0.051. Note the larger deviation in the curve.

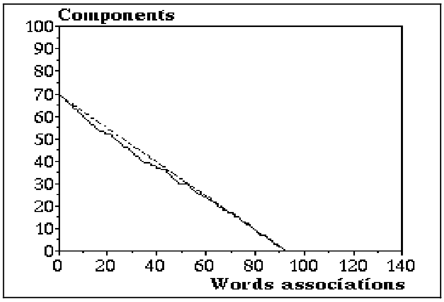


Fig. 3. Plot for a manual summary

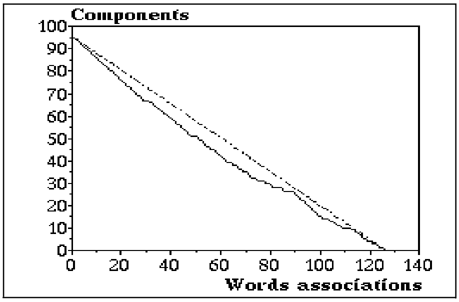


Fig. 4. Plot for an automatic summary

Antiqueira et al. performed their experiments with news texts, supposed to be good, and students' essays, supposed to be worse than the news texts. In this paper, we evaluate the possibility of adopting such method in summary evaluation. In order to do so, we first assume, as most works on summary evaluation do, that a summary must display the same properties a text presents in order to be classified as text. Therefore, summaries, as texts, must be coherent and cohesive, legible, grammatical, and present good thematic progression.

In our evaluation, we used a corpus called TeMário (Pardo and Rino, 2003) for Brazilian Portuguese. TeMário consists of 100 news texts from the on-line newspaper *Folha de São Paulo* (containing texts from Sections Special, World, Opinion, International, and Politics) and their corresponding manual summaries written by a professional abstractor. To our knowledge, TeMário is the only available corpus for summarization purposes for the Brazilian Portuguese language.

We compared the manual summaries to automatic summaries produced by 3 systems, namely, GistSumm (GIST SUMMarizer) (Pardo et al., 2003), SuPor

(SUmmarizer for PORtuguese) (Módolo, 2003) and GEI (*Gerador de Extratos Ideais*) (Pardo e Rino, 2004). We selected these systems for the following reasons: GistSumm is one of the first summarization systems publicly available for Portuguese; according to Rino et al. (2004), SuPor is the best summarization system for Portuguese; GEI was used to produce the automatic summaries that also accompany TeMário distribution. In what follows, each system is briefly explained. Then, our experiment is described and the results discussed.

4.1 Systems Description

The summarizers used in the evaluation are all extractive summarizers, i.e., they build the summary of a source text by juxtaposing complete sentences from the text, without modifying them. The summaries produced in this way are also called extracts.

GistSumm is an automatic summarizer based on a summarization method called gist-based method. It comprises three main processes: text segmentation, sentence ranking, and summary production. Sentence ranking is based on the keywords method (Luhn, 1958): it scores each sentence of the source text by summing up the frequency of its words and the gist sentence is chosen as the one with the highest score. Summary production focuses on selecting other sentences from the source text to include in the summary, based on: (a) gist correlation and (b) relevance to the overall content of the source text. Criterion (a) is fulfilled by simply verifying co-occurring words in the candidate sentences and the gist sentence, ensuring lexical cohesion. Criterion (b) is fulfilled by sentences whose score is above a threshold, computed as the average of all the sentence scores, to guarantee that only relevant sentences are chosen. All the selected sentences above the threshold are juxtaposed to compose the summary.

SuPor is a machine learning based summarization system and, therefore, has two distinct processes: training and extracting based on a Naïve-Bayes method, following Kupiec et al. (1995). It allows combining linguistic and non-linguistic features. In SuPor, relevant features for classification are (a) sentence length (minimum of 5 words); (b) words frequency; (c) signaling phrases; (d) sentence location in the texts; and (e) occurrence of nouns and proper nouns. As a result of training, a probabilistic distribution is produced, which entitles summarization in SuPor. In this paper, following Rino et al. (2004), we use the same features. SuPor works in the following way: firstly, the set of features of each sentence are extracted; secondly, for each of the sets, the Bayesian classifier provides the probability of the corresponding sentence being included in the summary. The most probable ones are selected to be in the summary.

Given a manual summary and its source text, GEI produces the corresponding ideal extract, i.e., a summary composed of complete sentences from the source text that correspond to the sentences content from the manual summary. This tool is based on the widely known vector space model and the cosine similarity measure (Salton and Buckley, 1988), and works as follows: 1) for each sentence in the manual summary, the most similar sentence in the source text is obtained through the cosine measure (based on word co-occurrence); 2) the most representative sentences are selected, yielding the corresponding ideal extract.

In general, ideal extracts are necessary to calculate automatically the amount of relevant information in automatic summaries produced by extractive methods. The automatic summaries are compared to the ideal extracts and two measures are usually computed: recall and precision. Recall is defined as the number of sentences from the ideal extract included in the automatic summary over the number of sentences in the ideal extract; precision is defined as the number of sentences from the ideal extract included in the automatic summary over the number of sentences in the automatic summary. A third measure, called f-measure, is a combination of recall and precision, being a unique measure of a summarization system performance.

As described by Rino et al. (2004), GistSumm and Supor participated in a comparative evaluation. Recall, precision and f-measure were computed for TeMário corpus, using the ideal extracts produced by GEI. A 30% compression rate was used in producing the automatic summaries. The compression rate specifies the size of the summary to be produced in relation to the source text in terms of number of words. In this case, the 30% compression rate specifies that the summary must have at most 30% of the number of words in the source text. Recall, precision and f-measure for GistSumm and SuPor are shown in Table 1, which reproduces part of the evaluation that Rino et al presented.

Table 1. Systems performance (in %)

Systems	Recall	Precision	F-measure
SuPor	40.8	44.9	42.8
GistSumm	25.6	49.9	33.8

As can be noted, GistSumm had the highest precision, but the lowest recall. SuPor presented the best f-measure, being, therefore, the best system. These results will be commented upon in the next subsection, which describes the complex network experiment conducted in this paper.

4.2 Experiment

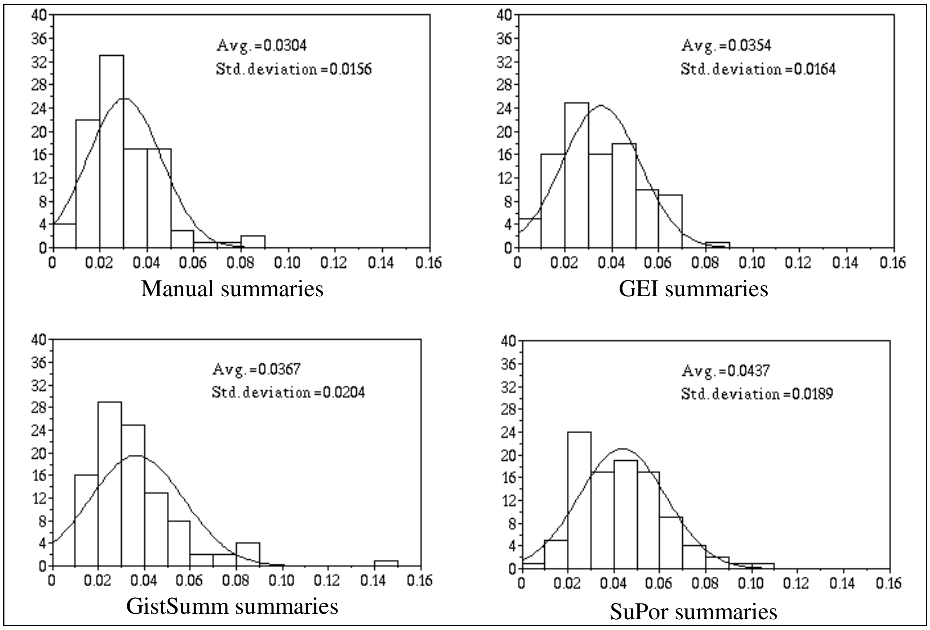
For running our experiment, we took the manual summaries and the ideal extracts (produced by GEI) that accompany TeMário and the corresponding automatic summaries produced by GistSumm and SuPor. As in Rino et al. (2004), we used a 30% compression rate. Based on our knowledge about the way the summaries were produced and on the evaluation that Rino et al. presented, we assume that the manual summaries are better than the ideal extracts, which are better than the automatic summaries. In terms of complex networks, the deviation from a straight line in the dynamics of network growth should be lowest for the manual summaries, and then increase for the ideal extracts and even more for the automatic summaries.

At this point, it is hard to predict how SuPor and GistSumm summaries will behave in relation to each other. Although SuPor is better than GistSumm in informativity evaluation (see Table 1), i.e., the amount of relevant information the summaries have, it is unlikely this will be reflected in the way we model summaries as complex networks. In fact, in the text quality experiment, Antiqueira et al. (2005a, 2005b) suggested that what is being captured by the complex network is the flow of new concepts introduced during the text: bad texts would introduce most of the concepts abruptly; good texts, on the other hand, would do it gradual and uniformly during the text development, resulting in a more understandable and readable text.

Table 2 shows the average deviation for each group of summaries and its increase in relation to the manual summaries deviation. For instance, for GistSumm (line 3 in the table), the average of the summaries deviation is 0.03673, which is 20.62% larger than the average deviation for the manual summaries.

Table 2. Experiment results

	Avg. deviation	Over manual summaries (%)
Manual summaries	0.03045	0
GEI	0.03538	16.19
GistSumm	0.03673	20.62
SuPor	0.04373	43.61



Manual summaries

GEI summaries

GistSumm summaries

SuPor summaries

Fig. 5. Histograms for summaries and their deviations

Using t-student test (Casella and Berger, 2001) for comparing the average deviations of our data, with 99% confidence interval, the p-values are below 0.03, which indicates that the resulting numbers are not due to mere chance. In other words, the results are statistically significant. The only exception was the p-value for the comparison between GistSumm and GEI, which was around 0.60. This happened because of the short distance between the results of the two systems, as Table 2 illustrates.

Figure 5 shows the histograms for the summaries and their respective deviations, where the x-axis represents the deviation and the y-axis the number of texts. As the average deviation grows for each group of summaries, the Gaussian distribution has its peak (which corresponds to the mean) displaced to the right, i.e. there are more texts with higher deviations.

As expected, the results suggest that manual summaries are better than the ideal extracts, and that these are better than the automatic summaries. This observation positively answers our question about the possibility of using complex networks to evaluate summaries in a comparative fashion. We claim that it must be restricted to a comparative evaluation because it is difficult to judge the validity of a deviation number without any reference. The results also show that, in contrast to the informativity evaluation, GistSumm outperformed SuPor in this experiment, as mentioned above as a possible result. We believe the reason for this to be the summarization method used by GistSumm: to produce the summary, it selects sentences that correlate with the gist sentence, resulting in a summary with similar thematic elements across the sentences and, therefore, with a more natural flow of concepts. With GistSumm and SuPor numbers, it is also possible to conclude for the truth of the assumption that our modeling of summaries as complex networks probably does not capture summary informativity or that alternative complex networks measurements may be necessary.

5 Conclusions

This paper presented an application of the approach described by Antiqueira et al. (2005a, 2005b) to summary evaluation, which is considered a hard problem in NLP. By modeling summaries as complex networks and by exploring a network metric, we showed it to be possible to distinguish summaries according to their quality. The evaluation presented here can be used in association to other automatic evaluations, complementing the results obtained with the traditional informativity metrics – recall and precision – or new ones – ROUGE (Lin and Hovy, 2003), for instance. Because it is based on abstract representation of texts in terms of complex networks, the proposed solution looks elegant, generic and language independent.

In the future, we plan to apply such evaluation to other text genres, in addition to the news texts. We also aim at investigating other network properties and their usefulness for characterizing the several aspects of a summary that is worth modeling and evaluating, e.g., coherence and cohesion. Other ways of modeling summaries as complex networks may also be explored.

Acknowledgments

The authors are grateful to CNPq and FAPESP.

References

Antiqueira, L.; Nunes, M.G.V.; Oliveira Jr., O.N.; Costa, L.F. (2005a). Modelando Textos como Redes Complexas. In *Anais do III Workshop em Tecnologia da Informação e da Linguagem Humana* – TIL. São Leopoldo-RS, Brazil. July 22-26.

Antiqueira, L.; Nunes, M.G.V.; Oliveira Jr., O.N.; Costa, L.F. (2005b). *Complex networks in the assessment of text quality*. physics/0504033.

Barabási, A.L. (2003), *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume, New York.

Cancho, R. F.; Capocci, A.; Caldarelli, G. (2005). *Spectral methods cluster words of the same class in a syntactic dependency network*. cond-mat/0504165.

Casella, J. and Berger, R.L. (2001). *Statistical Inference*. Duxbury, Belmont, California.

Costa, L.F. (2004). What's in a name? *International Journal of Modern Physics C*, Vol. 15, pp. 371-379.

Dorogovtsev, S.N. and Mendes, J.F.F. (2002). Evolution of networks. *Advances in Physics*, Vol. 51, N. 4, pp. 1079-1187.

Erkan, G. and Radev, D.R. (2004). Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research* – JAIR, Vol. 22, pp. 457-479.

Kupiec, J.; Pedersen, J.; Chen, F. (1995). A trainable document summarizer. In the *Proceedings of the 18th ACM-SIGIR Conference on Research & Development in Information Retrieval*, pp. 68-73.

Lin, C-Y. and Hovy, E.H. (2003). Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In the *Proceedings of Language Technology Conference* – HLT. Edmonton, Canada. May 27 - June 1.

Luhn, H. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, Vol. 2, pp. 159-165.

Mani, I. (2001). *Automatic Summarization*. John Benjamin's Publishing Company.

Mihalcea, R. (2005). Language Independent Extractive Summarization. In the *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, Michigan.

Miller, G.A. (1985). Wordnet: a dictionary browser. In the *Proceedings of the First International Conference on Information in Data*. University of Waterloo.

Módolo, M. (2003). *SuPor: um Ambiente para a Exploração de Métodos Extrativos para a Sumarização Automática de Textos em Português*. Master thesis. Departamento de Computação, UFSCar.

Motter, A.E.; Moura, A.P.S.; Lai, Y.C.; Dasgupta, P. (2002). Topology of the conceptual network of language. *Physical Review E*, Vol. 65, 065102.

Newman, M.E.J. (2003). The structure and function of complex networks. *SIAM Review*, Vol. 45, pp. 167-256.

Pardo, T.A.S. and Rino, L.H.M. (2003). *TeMário: Um Corpus para Sumarização Automática de Textos*. NILC technical report. NILC-TR-03-09. São Carlos-SP, October, 13p.

Pardo, T.A.S. and Rino, L.H.M. (2004). *Descrição do GEI - Gerador de Extratos Ideais para o Português do Brasil*. NILC technical report. NILC-TR-04-07. São Carlos-SP, August, 10p.

Pardo, T.A.S.; Rino, L.H.M.; Nunes, M.G.V. (2003). GistSumm: A Summarization Tool Based on a New Extractive Method. In N.J. Mamede, J. Baptista, I. Trancoso, M.G.V. Nunes (eds.), *6th Workshop on Computational Processing of the Portuguese Language - Written and Spoken* – PROPOR (Lecture Notes in Artificial Intelligence 2721), pp. 210-218. Faro, Portugal. June 26-27.

Rino, L.H.M.; Pardo, T.A.S.; Silla Jr., C.N.; Kaestner, C.A.; Pombo, M. (2004). A Comparison of Automatic Summarization Systems for Brazilian Portuguese Texts. In the *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence* – SBIA (Lecture Notes in Artificial Intelligence 3171), pp. 235-244. São Luis-MA, Brazil. September, 29 - October, 1.

Santos Jr. E.; Mohamed, A.A.; Zhao Q. (2004). Automatic Evaluation of Summaries Using Document Graphs. In the *Proceedings of the Workshop on Text Summarization Branches Out*, pp. 66-73.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, Vol. 24, pp. 513-523.

Sigman, M. and Cecchi, G.A. (2002). Global Organization of the Wordnet Lexicon. In the *Proceedings of the National Academy of Sciences*, Vol. 99, pp. 1742-1747.

SABio: An Automatic Portuguese Text Summarizer Through Artificial Neural Networks in a More Biologically Plausible Model

Télvio Orrú[1], João Luís Garcia Rosa[2], and Márcio Luiz de Andrade Netto[1]

[1] Computer Engineering and Industrial Automation Department,
State University of Campinas - Unicamp, Campinas, São Paulo, Brazil
{telvio, marcio}@dca.fee.unicamp.br
[2] Computer Engineering Faculty - Ceatec,
Pontifical Catholic University of Campinas - PUC-Campinas,
Campinas, São Paulo, Brazil
joaoluis@puc-campinas.edu.br

Abstract. An implementation of a computational tool to generate new summaries from new source texts in Portuguese language, by means of connectionist approach (artificial neural networks) is presented. Among other contributions that this work intends to bring to natural language processing research, the employment of more biologically plausible connectionist architecture and training for automatic summarization is emphasized. The choice relies on the expectation that it may lead to an increase in computational efficiency when compared to the so-called biologically implausible algorithms.

1 Introduction

The wave of information and the lack of time to read long texts make the modern society to look for abstracts and news headlines instead of complete texts. Towards an automatic text summarization system for Portuguese language, the system SABio (A*utomatic* S*ummarizer for the Portuguese language with more* Bio*logically plausible connectionist architecture and learning*) - a connectionist system employing more biologically plausible training algorithm and architecture - is proposed. An unique architecture for this application is presented, considering important features such as: (a) neural network training with more biologically plausible treatment; (b) training set (input-output pairs) is formed by features of source texts and ideal extracts, represented by elements of a multi-valued logic.

In order to present the system SABio, this paper is organized in the following sections: 1 - *Introduction*: this section; 2 - *Automatic Summarization*: brings fundamental concepts about text summarization; 3 - SABio: the proposed system is presented; 4 - *Evaluation of* SABio: comparisons between biologically plausible and implausible connectionist training algorithms for SABio are highlighted; and 5 - *Conclusion*.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 11–20, 2006.

2 Automatic Summarization

Text summarization is the process of production of a shorter version of a source text [1]. It is possible, through this process, to obtain an extract or an abstract. *Extracts* are created by the juxtaposition of source text sentences considered relevant; while *abstracts* alter the structure and/or content of original sentences, merging them and/or rewriting them, to generalize or specify information [2].

There are two points of view of a summary: the reader's (summary user) and the writer's (summary creator). The latter has the task of condensing a source text in a way that it is possible to transmit its main idea with the created summary.

From computational standpoint, three basic operations can describe the summarization process: analysis, content selection, and generalization [3]. An automatic system capable to make a condensation[1], with the preservation of the more relevant content of the source text, can be called a system for automatic text summarization.

An Artificial Neural Network (ANN) [4] is employed in the proposed system. Many systems use this computational tool, e.g.:

- Pardo et al. [5], in *NeuralSumm*, use an ANN of type SOM (self-organizing map), trained to classify sentences from a source text according to their importance degree[2] and then produce the related extract. The SOM network organizes information into similarity clusters based on presented features;
- Aretoulaki [3] uses an ANN with training considered biologically implausible (through supervised algorithm Back-propagation), differently from the approach proposed here (ANN with training considered more biologically plausible). Aretoulaki makes a detailed analysis of journalistic and scientific texts from several domains to recognize satisfactorily generic features that represent the sentence content of any textual type and that can be learned by a neural network. He considers features from several sources, from superficial to pragmatic, that can be identified in a superficial textual analysis.

There are automatic text summarizers that employ techniques to discover the more important sentences in source text [6, 5]. Such approaches are, in general, statistical because they try to organize sentences according to the frequency of their words in the text they belong. The sentences containing the more frequent words are called *gist sentences* and express the text main idea.

In order to evaluate automatic text summarizers, some rates are considered:

- *Precision Rate*: the amount of correctly selected sentences divided by the total amount of selected sentences;

[1] *Condensation* is the act of making something shorter (the *abstract*), according to Longman Dictionary.

[2] It is believed that sentences containing more frequent terms in source texts could present a greater importance in text.

- *Recall Rate*: the amount of correctly selected sentences divided by the total amount of correct sentences;
- *F-Measure*: two times the product of precision rate and recall rate divided by the sum of precision rate and recall rate [7].

Notice that the bigger the F-Measure, the better the generated extract, since it takes into consideration both recall and precision rates.

3 SABio

There are several implemented systems that propose automatic text summarization employing available corpora. These systems use symbolic as well as connectionist approaches. Implementations employing ANNs with training considered biologically implausible are often found, differently from this proposal.

Here, the system SABio (A*utomatic* S*ummarizer for the Portuguese language with more* Bio*logically plausible connectionist architecture and learning*) is proposed. The system produces extracts through a more biologically plausible training with ANNs.

The SABio proposal is partly motivated by the increasing interest of modern society in search of newspaper and magazines headlines instead of complete texts, mainly because of the lack of time of people nowadays.

3.1 The Biological Plausibility

According to O'Reilly and Munakata [8], there are evidences that the cerebral cortex is connected in a bi-directional way and distributed representations prevail in it. So, more biologically plausible connectionist (ANN) models should present some of the following characteristics:

- *Distributed representation*: generalization and reduction of the network size can be obtained if the adopted representation is distributed (several units for one concept, and similar concepts sharing units), since connections among units are able to support a large number of different patterns and create new concepts without allocation of new hardware;
- *Inhibitory competition*: the neurons that are next to the "winner" receive a negative stimulus, this way strengthening the winner neuron. In the nervous system, during a lateral inhibition, a neuron excites an inhibitory interneuron that makes a feed-back connection on the first neuron [9];
- *Bi-directional activation propagation*: the hidden layers receive stimuli from input and output layers. The bi-directionality of the architecture is necessary to simulate a biological electrical synapse, that can be bi-directional [9, 10];
- *Error-driven task learning*: in algorithm GeneRec - Generic Recirculation [11], the error is calculated from the local difference in synapses, based on neurophysiological properties, different from algorithm Error Back-propagation, which requires the back-propagation of error signals [12].

3.2 GeneRec: A Training Algorithm Considered More Biologically Plausible

The algorithm GeneRec - Generic Recirculation - was developed by O'Reilly [11] based on Back-propagation but considering properties of a more biologically plausible artificial neural network training algorithm.

GeneRec employs two phases: "minus" and "plus":

- *Minus Phase*: When units are presented to the input layer there is the propagation of this stimulus to hidden layer (bottom-up propagation). At the same time, the previous output propagates from the output layer to the hidden layer (top-down propagation). Then the "minus" hidden activation is generated (sum of bottom-up and top-down propagations). Finally, the real output is generated through the propagation of the "minus" hidden activation to the output layer. Notice that the architecture is bi-directional.
- *Plus Phase*: Units are presented again to the input layer; there is the propagation of this stimulus to hidden layer (bottom-up propagation). At the same time, the desired output propagates from the output layer to the hidden layer (top-down propagation). Then the "plus" hidden activation is generated, summing bottom-up and top-down propagations [11].

In order to make learning possible, the synaptic weights are updated, based on "minus" and "plus" hidden activations, real and desired outputs, input, and the learning rate [11, 4].

3.3 SABio **Features**

SABio's artificial neural network was trained with sentences[3] from a corpus called *TeMário*[4] that contains 100 journalistic texts in Portuguese language, with 61,412 words. 60 texts belong to the on-line Brazilian newspaper *Folha de São Paulo*[5], and the remaining 40 texts were published in *Jornal do Brasil* newspaper[6], also on-line version. These texts are distributed amongst distinct domains: opinions, critiques, world, politics, and foreign affairs [13].

The corpus *TeMário* is composed of source texts, manual summaries (abstracts), and ideal extracts. Manual summaries are produced by the authors of the source texts, after a rewritten process of the content "judged" more relevant.

Manual summaries are not employed in SABio because: (*a*) there is no explicit correspondence between source texts and manual summaries; and (*b*) their production is costly and long-lasting, since it requires human beings. Instead, SABio uses the ideal extracts[7] available in corpus *TeMário*. They are automatically produced by a system called GEI through the employment of the

[3] Two thirds of texts were used for training and one third for testing.

[4] http://www.nilc.icmc.usp.br/nilc/tools/corpora.htm

[5] http://www.folha.uol.com.br/

[6] http://jbonline.terra.com.br/

[7] Here, *ideal summaries* and *ideal extracts* are considered the same, although there are differences between them.

cosine measure [14]: for each sentence of the manual summary, the correspondent sentence in the most similar source text is searched. This is done by word co-occurrence: the larger the number of words from the manual summary a source text sentence has, the greater its chance of presenting the same content of the summary sentence. This way, it could be employed to compose the ideal summary [5].

SABio presents satisfactory results, although it has employed only lexical processing and cue words, and not dealt with other linguistic information, such as syntactic and semantic analysis, in order to develop a summary.

TeMário was chosen among several corpora, because it would be easier to make comparisons in the future with other summarizers that also employ this corpus. It was considered also that international automatic summarizer evaluations, like SUMMAC - text SUMMArization evaluation Conference - and DUC - Document Understanding Conference - have employed journalistic texts.

In training step, for each source text sentence[8], seven features that represent the input sentences are analysed. These coded sentences are associated to a desired output, which is classified according to the importance degree of the sentence in relation to the ideal extract. The possible values are: *None*, *Small*, *Small-Medium*, *Medium*, *Medium-Large*, and *Large* frequency. The classification of the frequency for the sentences that represent the desired outputs is obtained through the method of the *gist sentence*[9] discovery.

For every sentence all word letters are converted to upper case, for uniformity [15]. The employed features are [5]:

1. *Size of the sentence*: long sentences often present greater informative content, considered more relevant for the text [16]. The size of a sentence is calculated taking into account the amount of words that belong to it, except the words belonging to the StopList[10]. The sentences in the text are classified as: *Small*, which represents the size of the smaller sentence in the text; *Medium*, representing the average-size sentence, or *Large*, which represents the size of the larger sentence in the text;
2. *Sentence position in text*: the position of the sentence can indicate its relevance [3]. In SABio, similar to NeuralSumm [5], it is considered that a sentence can be in the beginning (first paragraph), at the end (last paragraph), or in the middle (remaining paragraphs) of the text;
3. *Sentence position in paragraph where it belongs*: the position of the sentence in the paragraph can also indicates its relevance [18]. In SABio, it is considered that a sentence can be in the beginning (first sentence), at the end (last sentence), or in the middle (remaining sentences) of the paragraph;

[8] In SABio the end of a sentence can be indicated by conventional punctuation marks: period, exclamation mark, or question mark.

[9] To know which is the *gist sentence*, the approach mentioned in GistSumm [6] is used. In this approach, a sentence is positioned according to the importance degree it represents in the text where it belongs.

[10] *StopList* is a list of very common words or words considered irrelevant to a text, mainly because of their insignificant semantic values [17].

4. *Presence of gist sentence words in the sentence*: sentences that contain words of the gist sentence, that is, a sentence that better expresses the text main idea, tends to be relevant [6];
5. *Sentence value based on the distribution of words in the text*: sentences with high value often are relevant to the text [19]. The value of each sentence is calculated by the sum of the occurrence number of each one of its words in the whole text divided by the number of the words in the sentence, and the obtained result in this operation will be related to the values mentioned in the first feature (Small, Medium, and Large);
6. *TF-ISF of the sentence*: sentences with high value of TF-ISF (Term Frequency - Inverse Sentence Frequency) are representative sentences of the text [20]. For each word of a sentence, the TF-ISF measure is calculated by the formula:

   $$TF{-}ISF(w) = F(w) \times \frac{log(n)}{S(w)}$$

   where
   F(w) is the frequency of the word *w* in the sentence,
   n is the number of words in sentence in which *w* belongs, and
   S(w) is the number of sentences in which *w* appears.
   The TF-ISF value of a sentence is the average of the TF-ISF values of each one of its words, and the obtained result of this equation will be related to the values mentioned in first feature (Small, Medium, and Large);
7. *Presence of indicative words in the sentence*: indicative words (cue words) often indicate the importance of the sentence content [21]. This feature is the only dependent on language, genre, and text domain. SABio uses the same indicative words used in NeuralSumm [5]: evaluation, conclusion, method, objective, problem, purpose, result, situation, and solution.

4 Evaluation of SABio

The comparisons between the training algorithms Back-propagation and GeneRec for SABio were conducted concerning the number of epochs necessary for convergence, processing time, and quality of the generated summary.

The first comparison aims to show the necessary time for the ANN convergence with the training algorithms Error Back-propagation [12] and GeneRec [11]. In order to achieve this, tests were conducted with learning rates 0.10, 0.25, 0.35, or 0.45 and also tests with different numbers of neurons in hidden layer: 10, 11, 12, 20, or 25.

It was observed that when SABio's ANN was trained with GeneRec, the minimum error[11] could be reached within a smaller number of epochs, when compared to Back-propagation. Figure 1 shows the best performances for both algorithms.

Although the displayed results in Fig. 1 intend to compare minimum errors in relation to the number of epochs and processing time for the algorithms

[11] The mean quadratic error formula is related to the derivative of the activation function (sigmoid) [4]. For the comparisons, it was considered that the network "converged" when the minimum error reached the value of 0.001.

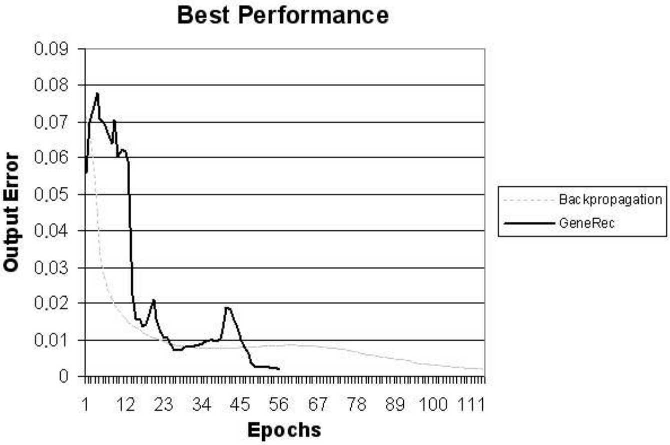


Fig. 1. Best performances in SABio tests. Back-propagation with 25 hidden neurons and learning rate of 0.35 converges in 115 epochs and 8 seconds. GeneRec with 11 hidden neurons and learning rate of 0.45 converges in 57 epochs and 3 seconds. The curve for GeneRec keeps its saw aspect for epochs over 57 (not shown in the figure). But this is irrelevant in this case, since the minimum error had already been reached.

GeneRec and Back-propagation, it is considered relevant to state that the recall rate obtained by the GeneRec training revealed greater than the recall rate by Back-propagation training (the F-Measure was respectively 33.27 and 30.97).

The comparison of SABio with other automatic text summarizers considered the recall and precision rates of summaries generated by SABio. To know which is the best architecture that SABio could employ[12] to reach greater recall and precision rates, preliminary tests with several architectures were conducted, altering the number of epochs, learning rates, and amount of hidden neurons[13]. The best architecture found in tests was used to compare SABio with other automatic text summarizers.

In order to make comparisons between SABio and other automatic summarizers, the F-Measure was employed because besides including recall and precision rates in its formula, it is often employed to compare the efficiency of automatic summarizers in relation to their generated summaries [22].

The same conditions found in Rino et al. [22] were employed, that is:

- a compression rate of 70%. Compression rates usually range from 5% to 30% of the source text content [2]. Other rates displaying similar performances were experimented;

[12] In order to make comparisons between treatments considered more biologically plausible and implausible through SABio architecture, adaptations were provided in SABio in order to be trained with Back-propagation.

[13] Preliminary tests were conducted with: a) learning rates: 0.05, 0.15, 0.25, and 0.45; b) epochs: 2,000 until 10,000 (multiples of 2,000); c) hidden neurons: 8, 16, and 22.

– a 10-fold cross validation, non-biasing process. TeMário was divided into ten distinct groups of texts so that each group contains ten different texts. For the training set, nine of these groups were employed and for the test, the remaining group. This way, the texts used for the test do not belong to the training set. Ten tests were performed (one for each group), and recall, precision, and F-Measure rates were calculated. Then, averages of these rates for the ten experiments were extracted.

This experiment made possible the comparison of SABio with other automatic text summarizers which employ exactly the same method and the same corpus. Table 1 shows the comparative frame among them.

Table 1. Performance (in %) of the systems: SABio-GR, trained by algorithm GeneRec, with 22 hidden neurons, learning rate of 0.25 and 4,000 epochs (501 seconds), SABio-EB, trained by algorithm Error Back-propagation, with 16 hidden neurons, learning rate of 0.45 and 8,000 epochs (896 seconds), among other automatic text summarizers. Adapted from Rino et al. [22].

Summarizer	*Precision rate*	*Recall rate*	*F-Measure*	*Difference to SABio-GR's F-Measure in %*
Supor	44.9	40.8	42.8	1.90
ClassSumm	45.6	39.7	42.4	0.95
SABio-GR	**43.8**	**40.3**	**42.0**	**-**
SABio-EB	**42.4**	**38.7**	**40.5**	**-3.70**
From-Top	42.9	32.6	37.0	-13.51
TF-ISF-Summ	39.6	34.3	36.8	-14.13
GistSumm	49.9	25.6	33.8	-24.26
NeuralSumm	36.0	29.5	32.4	-29.63
Random order	34.0	28.5	31.0	-35.48

SABio outcomes can be considered satisfactory, because:

– the first place - *Supor - Text Summarization in Portuguese* [23] reached performance[14] 1.90% above SABio-GR, but it employs techniques that make computational cost higher than SABio, like lexical chains and thesaurus;
– the second place - *ClassSumm - Classification System* [24] - displayed performance 0.95% above SABio-GR, but it also employs high-cost techniques like semantic analysis, similarity of the sentence with the title (its ideal extracts must have titles), anaphor occurrence analysis, and tagging.

The third place of SABio-GR can be considered a very good performance. In the fourth place, SABio appears again, but with adaptations in order to run with algorithm Back-propagation (version EB).

[14] The F-Measure is used for performance measurement.

SABio presents some similarities with NeuralSumm [5] regarding the employed training set features (7 of 8 NeuralSumm features are present in SABio - the absent feature regards the presence of keywords in the sentence, whose relevance is questionable). The search for more adequate classifiers for automatic summarization is as important as the selection of the features that better represent the focused problem [24, 5].

5 Conclusion

The increasing interest in applications of automatic text summarizers can be justified by the need of the modern society in searching of headlines instead of complete texts in magazines and newspapers, mainly because of lack of time.

Presenting the system SABio, this paper aims to show that it is possible to achieve better performance in automatic summarization when a more biologically plausible model for the ANN training is employed. It is not intention to reveal weaknesses in existent automatic summarizers neither make comparisons that can state that one summarizer is “better” than another. Of course, this “choice” would rely on the “best” features chosen for an automatic summarizer.

SABio presents several limitations, and it can be improved, but the employment of more biologically plausible models for automatic text summarization could represent the achievement of better performance, with greater recall and precision rates, and also contribute to restore principles of ANNs.

References

1. Mani, I., Maybury, M.T.: Advances in Automatic Text Summarization. MIT Press, Cambridge, MA, USA (1999)
2. Mani, I.: Automatic Summarization. John Benjamins Pub., Amsterdam (2001)
3. Aretoulaki, M.: COSY-MATS: A Hybrid Connectionist-Symbolic Approach to the Pragmatic Analysis of Texts for Their Automatic Summarisation. PhD thesis, Univ. of Manchester (1996)
4. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2 edn. Prentice Hall, Upper Saddle River, New Jersey, USA (1999)
5. Pardo, T.A.S., Rino, L.H.M., Nunes, M.G.V.: NeuralSumm: Uma Abordagem Conexionista para a Sumarizacão Automática de Textos. In: Anais do IV Encontro Nacional de Inteligência Artificial - ENIA2003. Volume 1., Campinas, São Paulo, Brazil (2003) 1–10
6. Pardo, T.A.S., Rino, L.H.M., Nunes, M.G.V.: GistSumm: A Summarization Tool Based on a New Extractive Method. 6th Workshop on Computational Processing of the Portuguese Language **6** (2002) 210–218
7. van Rijsbergen, C.J.: Information Retrieval. 2 edn. Butterworth, London, England (1979)
8. O’Reilly, R.C., Munakata, Y.: Computational Explorations in Cognitive Neuroscience - Understanding the Mind by Simulating the Brain. A Bradford Book, The MIT Press, Cambridge, Massachusetts, USA (2000)
9. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Essential of Neural Science and Behavior. Appleton and Lange, Stanford, Connecticut, USA (1995)

10. Rosa, J.L.G.: A Biologically Inspired Connectionist System for Natural Language Processing. In: Proc. of the 2002 VII Brazilian Symposium on Neural Networks - SBRN2002, Recife, Brazil, IEEE Computer Society Press (2002) 243–248
11. O'Reilly, R.C.: Biologically Plausible Error-driven Learning using Local Activation Differences: The Generalized Recirculation Algorithm. Neural Computation **8:5** (1996) 895–938
12. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In Rumelhart, D.E., McClelland, J.L., eds.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1. A Bradford Book, MIT Press (1986) 318–362
13. Pardo, T.A.S., Rino, L.H.M.: TeMário: Um Corpus para Sumarizacão Automática de Textos. Technical report, NILCTR-03-09 - ICMC-USP, São Carlos, São Paulo, Brazil (2003)
14. Salton, G.: Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley (1989)
15. Witten, I.H., Moffat, A., Bell, T.C.: Managing Gigabytes. Van Nostrand Reinhold, New York, NY, USA (1994)
16. Kupiec, J., Pedersen, J., Chen, F.: A Trainable Document Summarizer. ACM SIGIR **1** (1995) 68–73
17. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall (2000)
18. Baxendale, P.B.: Machine-Made Index for Technical Literature: An Experiment. IBM Journal of Research and Development **2** (1958) 354–365
19. Black, W.J., Johnson, F.C.: A Practical Evaluation of Two Rule-Based Automatic Abstraction Techniques. Expert Systems for Information Management **1(3)** (1988) 159–177
20. Larocca, J.N., Santos, A.D., Kaestner, C.A.A., Freitas, A.A.: Generating Text Summaries through the Relative Importance of Topics. In Monard, M.C., Sichman, J.S., eds.: Lecture Notes in Computer Science - Advances in Artificial Intelligence, Proc. of the Intl. Joint Conf. 7th. Ibero-American Conf. on AI - 15th. Brazilian Symposium on AI - IBERAMIA-SBIA 2000. Volume 1952., São Paulo, Brazil, Springer-Verlag Heidelberg (2000) 301–309
21. Paice, C.D.: The Automatic Generation of Literature Abstracts: An Approach Based on the Identification of Self-indicating Phrases. In: Proc. of the 3rd annual ACM Conf. on Research and Development in Information Retrieval, Cambridge, England, Butterworth (1981) 172–191
22. Rino, L.H.M., Pardo, T.A.S., Jr, C.N.S., Kaestner, C.A.A., Pombo, M.: A Comparison of Automatic Summarization Systems for Brazilian Portuguese Texts. In: Proc. of the XVII Brazilian Symposium on Artificial Intelligence - SBIA2004, São Luís, Maranhão, Brazil (2004) 235–244
23. Modolo, M.: SUPOR: Um Ambiente para a Exploracão de Métodos Extrativos para a Sumarizacão Automática de Textos em Português. PhD thesis, Computing Department - UFSCar, São Carlos, São Paulo, Brazil (2003)
24. Larocca, J.N., Freitas, A.A., Kaestner, C.A.A.: Automatic Text Summarization Using a Machine Learning Approach. In: Proc. of the 16th Brazilian Symposium on Artificial Intelligence, Porto de Galinhas, Pernambuco, Brazil (2002) 205–215

Building a Dictionary of Anthroponyms

Jorge Baptista[1,2], Fernando Batista[1,3], and Nuno Mamede[1,4]

[1] L2F – Laboratório de Sistemas de Língua Falada - INESC ID Lisboa,
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
jbaptis@ualg.pt, {Fernando.Batista, Nuno.Mamede}@inesc-id.pt
http://www.l2f.inesc-id.pt/
[2] Universidade do Algarve, Faculdade de Ciências Humanas e Sociais,
Campus de Gambelas, P – 8005-139 Faro, Portugal
[3] ISCTE – Instituto de Ciências do Trabalho e da Empresa,
Av. Forças Armadas, 1649-026 Lisboa, Portugal
[4] Instituto Superior Técnico - Universidade técnica de Lisboa,
Av. Rovisco Pais, 1049-001 Lisboa

Abstract. This paper presents a methodology for building an electronic dictionary of anthroponyms of European Portuguese (DicPRO), which constitutes a useful resource for computational processing, due to the importance of names in the structuring of information in texts. The dictionary has been enriched with morphosyntactic and semantic information. It was then used in the specific task of capitalizing anthroponyms and other proper names on a corpus automatically produced by a broadcast news speech recognition system and manually corrected. The output of this system does not offer clues, such as capitalized words or punctuation. This task expects to contribute in rendering more readable the output of such system. The paper shows that, by combining lexical, contextual (positional) and statistical information, instead of only one of these strategies, better results can be achieved in this task.

1 Introduction

The recognition of proper names in texts is a recurring problem in different domains of Natural Language Processing (NLP) such as Information Retrieval, Information Extraction, Machine Translation, Syntactic Analysis, Named Entity Recognition and, for example, in actor's identification in the discourse for dialogue systems [1,2,3,4]. Conventionally, the identification of proper names in an untagged text is carried out by using a linguistic [1] or a probabilistic approach [3,5,6]. In each case, a dictionary containing information about this type of words may constitute an important NLP resource for automatic corpora processing [7].

It may strike one as unusual to create a dictionary of proper names because, as opposed to words of the common lexicon, they are regarded as constituting a (potentially) infinite set. However, certain subclasses can be closed. For example, most *first names* (or *given* or *Christian names*)[1] can be intuitively identified whereas *last names*

[1] This terminology is somewhat inadequate, when dealing with non-Christian cultures (e.g. Muslim, Hebraic) or with cultures where surnames precede given names (e.g. Chinese, Korean). Naming conventions vary according to culture but, for the purpose of this paper, we center on Portuguese naming conventions.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 21 – 30, 2006.

(or *surnames*, or *family names*) are more often ambiguous with words of the common lexicon. Several kinds of proper names can be considered: anthroponyms (names of persons) and institutions' names [8]; toponyms (names of locations), hidronyms (names of rivers, lakes, seas and oceans) and other geographical accidents (mountains, chains, isles, and so on) [1]; ergonyms (manufactured objects and products), commercial names; siglae and acronyms [8]; and several others. Each one of these subclasses poses specific representation and recognition problems, such as their internal structure, multilexical forms and lexical ambiguity.

This paper presents an electronic dictionary of anthroponyms (DicPRO), which constitutes a resource for computational processing of texts in European Portuguese. In the future, we aim at treating and including other proper names' subclasses, allowing for a broader use of the resource.

The paper is structured as follows: the next section resumes the main linguistic features that can be associated to anthroponyms. Section 3 presents the methodology applied in the building of the DicPRO. Section 4 describes an experiment to assess the usefulness of this resource, in the specific task of capitalization of proper names. This was done by applying the dictionary to a corpus of broadcast news speech recognition system [9], automatically produced and then manually corrected. Usually, the output of the system does not offer formal clues, such as capitalized words and punctuation, which are otherwise used to identify proper names. We then compare this strategy with the alternative approaches of using only probabilistic methods for capitalizing proper names, using contextual (positional) rules based on lexical information made available by DicPRO and, finally, by combining together these strategies. The paper ends with some conclusions and perspectives for future work.

2 Anthroponyms

Anthroponomy is a subdomain of onomastics that deals mainly with the formation of personal names. Anthroponyms are of particular syntactic and semantic interest in linguistics, since they show special referential values and determination/modification constraints, and particular discourse values [10, 11, 12, 13]. This paper, however, will only focus on the most relevant linguistic features that should be encoded in an electronic dictionary of anthroponyms in view of their automatic recognition in texts. Furthermore, our approach will mainly consider the recognition of anthroponyms based on internal evidence [3], that is, without reference to contextual clues. We will see, however, that contextual (positional) and orthographic/probabilistic information can be used *in combination* to improve results in the specific task of capitalization that we have in mind.

As far as (European) Portuguese is concerned, it is possible to structure this class of names in two major subsets: *first names* and *last names*, each having different morphosyntactic properties. However, many names may be used both as first and last name: *Rosa*, even if one of the uses is often more frequent than the other.

First names can be marked with the gender property (*João*, masc./ *Sara*, fem.). In some cases they can present a diminutive form, which may be derived from the deletion of syllables: *Tó* (=*António*) or from affixation of diminutive suffixes: *Carlinhos*

(=*Carlos*) or even by combining both processes together: *Nelinha* (=*Manuela*). Last names do not have gender and usually do not admit the formation of diminutives. In this paper, however, diminutives were not considered.

The information about gender is relevant for the syntactic processing, for example, anaphora resolution, and should be taken into consideration when building a resource such as DicPRO. The information about diminutives may also be of pragmatic importance, since they may be used in several ways in discourse, to express affection, irony, familiarity, or other values.

Anthroponyms in texts only seldom show plural inflection. First names also differ from last names in this respect since they can often take a plural morpheme (usually the *–s* ending: *António*, sing./ *Antónios*, pl., cases like: *João*, sing./ ?*Joões*, pl., being rather rare and barely acceptable) while last names, even if they may (rarely) take plural endings: *o Silva* / *os Silvas* (the_sing. Silva / the_pl. Silva_*s*), they may also refer to a group of persons (usually a family) without any explicit marking (*os Silva*, the_pl. Silva). In this case, number of last name can only be ascertained from the preceding article. In the current state of the dictionary, plural was not considered.

The naming of a person is frequently made by combining several anthroponyms, eventually using connectors such as *e* (and) or *de* (of); preposition *de* can also be contracted with definite articles: *do*, *da*, *dos*, *das* (of+the) or, in special cases, be abbreviated as *d'* (of). These combinations are governed by culturally determined rules. Several aspects of the automatic syntactic analysis require the full identification of these sequences of proper names, in order to assign them the status of a named entity. However, this analysis may be error prone; for example, the conjunction may be ambiguous between a compound proper name (*Vale e Azevedo*, former president of a football club) or two coordinated noun phrases (*Soares e Cavaco*, two candidates for Presidency).

Finally, a significant percentage of DicPRO proper names (about 43%, mainly last names) are also ambiguous with words of the common lexicon, e.g. *Figo*. This ambiguity leads to an additional difficulty in recognizing the sequence used in naming a person, but it can be (at least partially) solved by means of disambiguating local grammars [6, 14].

3 Building the Dictionary

The anthroponyms dictionary was constructed semi-automatically, by selecting word forms from institutional official lists of proper names. This approach consisted of selecting words from two different types of sources: (a) a **List1** of isolated words collected from the phone directory of a Portuguese telephone company [15], and (b) a **List2** of complete person names, collected from lists of university students.

Each word form in List1 included some additional information such as: (i) its frequency in the entire phone directory and the indication of its occurrence as a first (F) or as a last (L) name. From the approximately 10,000 different forms having frequency higher than 10, a manual selection was made, resulting in about 4,500 anthroponyms. The classification of each name as first or last was then manually verified and (eventually) corrected, and first names were given their gender values.

List2, containing approximately 8,100 complete person names, was first processed manually, by establishing the border between first (= given) names and last names (= surnames). This first step allowed us to unequivocally determine the frequency of use of each word as first or/and last name. To all new names, that is names not yet present in the first list, gender, as well as other information was added, in particular, indication of foreign names: *Yuri*, and orthographic variants: *Melo* vs. *Mello*.

The resulting dictionary integrates, in its current state, approximately 6,200 different entries. In addition to the previously described process, each word in the dictionary was checked against a lexical resource of common lexicon entries [16], and all ambiguous words were marked (Amb). Table 1 characterizes the current content of DicPRO.

Table 1. DicPro content

Total Entries:	6,173	
from List1:	4,533	73.5 %
from List2: (not in List1)	1,640	26.5 %
(already in List1)	1,756	
F: first names (only):	1,870	30.3 %
L: last names (only):	4,200	68.0 %
names both F and L:	103	1.7 %
ambiguous :	2,629	42.6 %
ambiguous F	468	7.0 %
ambiguous L	2,196	35.0 %
ambiguous both F and L	35	0.6 %

The entries of DicPRO were formatted using the DELA formalism for INTEX [14]. Some entries are shown below:

```
Alegre,Alegre.N+Hum+Npr+CAP+L+Amb+L1
Gil,Gil.N+Hum+Npr+CAP+L+F+L1:ms
Tó,António.N+Hum+Npr+Dim+CAP+F+Amb:ms
```

From the information encoded in the dictionary, it is possible to remark that last names constitute the majority of DicPRO entries, and thus this set is likely to be expanded. On the other hand, first names are likely to constitute a more close set, not easily subject to further expansion. The number of proper names functioning simultaneously as first and last names may be considered residual. Secondly, ambiguous anthroponyms constitute an important part of DicPRO, especially last names, half of which are ambiguous with common lexicon entries.

To deal with orthographic variation[2] of proper names (for example, the use of voiceless consonants: *Baptista/Batista*, *Victor/Vítor* or double consonants:

[2] In some cases, these are not exactly variants, but correspond to former (often etymologically derived) spellings of the name, in previous orthographic conventions of Portuguese, which may coexist with the new spellings. Some of these spellings, however, should not be considered 'orthographic' variants, but rather a new and somehow increasingly fashionable spelling of proper names.

Estrella/Estrela, *Mattos/Matos*; use of digraphs instead of simple consonants: *Sophia/Sofia*, *Athaíde/Ataíde*; use of *y* instead of *i*: *Heytor/Heitor*), a set of enhanced morphological finite-transducers (FSTs) were built using INTEX linguistic development platform. These FSTs take an input string (an orthographic variant of a proper name) and validate them against the DicPRO entries (lemmas), while undoing the formal variation that gave rise to the variant and thus producing the lemma. Fig. 1 illustrates some variations described by these FSTs.

At this stage, only simple words, i.e., one-word names have been included in the dictionary, thus ignoring several compound names (e.g. *Corte-Real*, *Mil-Homens*, *Sim–Sim*, *Vila-Real*), seldom appearing in texts.

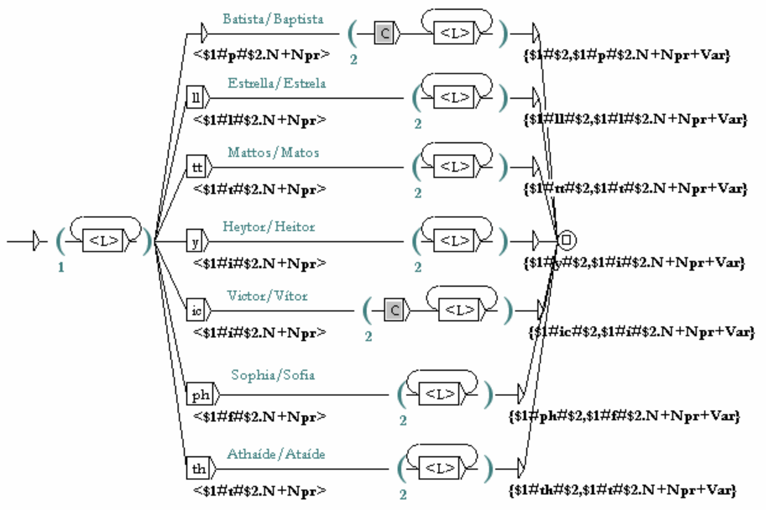


Fig. 1. A finite-state transducer to recognize and lemmatize orthographic variants of proper names

4 An Experiment on Capitalization

In order to assess the usefulness of the DicPRO resource, several experiments were carried out on the specific task of capitalization of proper names. Two subtasks were considered: *subtask 1* – only evaluates the capitalization of anthroponyms; *subtask 2* – evaluates the capitalization of all proper names, regardless of their anthroponymic status.

4.1 Methods

Corpus. The experience was carried out by applying the dictionary to a corpus of broadcast news speech recognition system [9], automatically produced and then manually corrected. Each proper name (anthroponyms and other) in the corpus has been capitalized and it was preceded by the sign '^'. Usually, the system's output does not offer any formal clues, such as capitalized words and punctuation, which are otherwise used to identify proper names. This fact results in a less then optimal reading quality of the output, that the capitalization task is intended to improve.

The corpus contains approximately half million words, and was divided in two subcorpora: (i) a *training corpus* with about 498,000 (27,513 different) words; and (ii) an *evaluation corpus* with 49,306 (7,513 different) words. Anthroponyms were then distinguished from other proper names in the evaluation corpus, by manually replacing the sign '^' by '#'. The following is a small sample of the evaluation corpus (anthroponyms and other proper names have been highlighted in bold)[3]:

> Jornal Dois, a informação com **#Manuel #Menezes**.
> Boa noite.
> A Comissão Europeia decidiu pedir a **^Portugal** que explique alguns aspectos do traçado da auto-estrada do **^Algarve**. Em causa está o projectado troco da ~A dois, que atravessa a zona de protecção especial de **^Castro ^Verde**, e que poderá constituir uma violação da directiva comunitária sobre protecção das aves selvagens.

The evaluation corpus contains 3,001 proper names, of which 1,101 are anthroponyms.

P(robabilistic)-Dic. In order to determine how much the DicPRO might improve the capitalization task of proper names, as compared with orthographic probability of a given word to be written in upper case, we produced a probabilistic dictionary from the training corpus. Each word of this corpus was assigned a capitalization probability depending on how many times it appeared in upper case (CAP) or in lower case (MIN) form. A word is given the CAP tag if it had >50% number of occurrences in capitals; else, the MIN tag was accorded. The list of word forms of the training corpus with this probabilistic information constitutes the P(robabilistic)-Dic. 15% of P-Dic forms were tagged with CAP. P-Dic covers about 83% of the word forms of the evaluation corpus. CAP or MIN information was also added to the DicPRO entries in order to enrich the resource. Table 2 resumes the content of P-Dic.

Table 2. P-Dic content

Total Entries:	25,528 [4]	
CAP:	3,822	15.0 %
MIN	21,706	85.0 %
also in DicPRO	1,373	
CAP	899	65.5 %
MIN	474	34.5 %

Local Grammars. A set of local grammars were built combining positional and lexical information[4]. These automata may be viewed as rules to recognize sequences of

[3] Notice that the proper name *Castro* in the compound toponym *Castro Verde* has not been given the anthroponyms tag (#), even if it can also be used as such.

[4] These local grammar were adapted from [17].

words (and eventual connectors), that are likely to be proper names and should therefore be spelled in upper case. In the scope of this paper, a set of 10 rules were considered. Every rule can be used in a standalone mode, however, the idea is to build an integrated grammar, combining all the separated rules, which will provide the best result. Fig. 2 illustrates two examples of these rules.

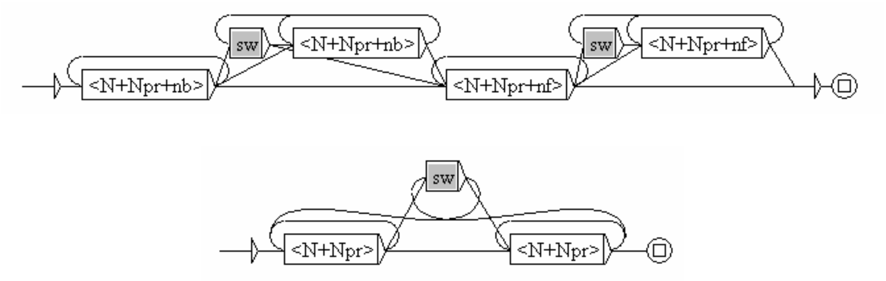


Fig. 2. Rules 8 and 9. Two finite-state automata to recognize sequences of words, candidates for proper names. The first rule (rule 8) identifies sequences of at least one first name: <N+Npr+F> followed by at least one last name: <N+Npr+F>; eventual connectors – but not conjunction e (and) – are represented by the auxiliary graph (grey box) sw (=stopword). The second rule (rule 9) is less specified: it identifies any sequence of proper names (and any eventual connectors), regardless of their F or L tags.

4.2 Results and Discussion

Several experiments were conducted in order to find the best way of identifying: anthroponyms and proper names in general. In these experiments, different methods of capitalization were compared, namely: a) using only the DicPRO information (experiment 1); b) using only probability information regarding the use of upper case in a training corpus (experiment 2); c) using the DicPRO with contextual (positional) information (experiments 3, 4 and 5); d) combining the different methods (experiments 6 and 7). Results are shown in Table 3.

Table 3. Results obtained from 7 different experiments on identifying anthroponyms and proper names. F = F-Measure, MaxF = F-measure for the two subtasks' F-measures

	subtask 1 (anthroponyms)			subtask 2 (proper names)			
Experiment	Prec.	Recall	F	Prec.	Recall	F	Max F
1. <N+Npr>	32,6%	79,3%	0,462	60,1%	53,6%	0,566	0,509
2. <WORD+CAP>	30,3%	79,9%	0,439	72,7%	70,4%	0,715	0,544
3. Rule 8	86,6%	51,0%	0,641	97,3%	20,5%	0,338	0,443
4. Rule 9	70,5%	65,5%	0,679	92,2%	31,4%	0,468	0,554
5. Rules 1-9	63,6%	74,8%	0,687	93,6%	40,4%	0,564	0,619
6. Rule 10	58,4%	69,5%	0,634	89,2%	39,0%	0,542	0,585
7. Rules 1-10	30,5%	87,0%	0,451	71,8%	75,2%	0,734	0,559

The first two experiments help define precision and recall baseline values for the two main methods of capitalization, namely the separate use of DicPRO against the separate use of P-Dic. For subtask 1, their results are approximately equivalent, even if the P-Dic shows a slightly better F-measure. Overall, a similar baseline precision of ±30% and ±80% recall can be expected for both methods in this subtask. However, for subtask 2, as expected, P-Dic achieves much better results since the lexical coverage of DicPRO is limited to anthroponyms.

Experiments 3, 4 and 5 illustrate the combined use of DicPRO with several contextual (positional) rules; here the separate results of the best performing rules (rules 8 and 9, shown in Fig. 2) are given, while experiment 5 shows the result of the combination of all rules. It is clear that, for subtask 1, the combined use of contextual rules and the DicPRO achieves much better results, compared to the two baseline experiments. Previous rules (rules 1 to 7), not shown here, were highly specific having very high precision but very low recall. Each rule seems to capture different combinatorial phenomena, but experiment 5, which is the conjunction of all rules, achieves a better F-measure. For subtask 2, these experiments achieved the best precision results, at the cost of lowering recall. Nevertheless, F-measure of experiment 5 is only slightly less that that of experiment 1. Still, one should bear in mind that DicPRO information only regards anthroponyms, thus a low recall on subtask 2 should be expected.

Experiment 6 combines the three methods, by using rule 10, shown in Fig. 3.

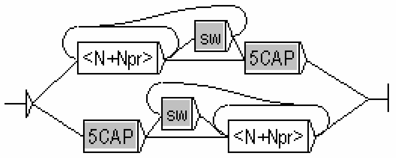


Fig. 3. Rule 10 - Combining DicPRO, P-Dic and contextual rules. A finite-state automata to recognize sequences of (up to 5) words not included in DicPRO but otherwise marked as CAP on the P-Dic, followed or preceded by anthroponyms.

In both subtasks 1 and 2, adding probabilistic information does seem to improve results significantly, as compared to experiments 3 to 5. However, it outperforms precision of both experiments 1 and 2 in subtask 1, while showing a lower recall.

Experiment 7, combines experiments 2, 5 and 6. In both subtasks 1 and 2, this experiment attains the best performance in recall (87% and 75%, respectively), even if precision is equivalent to the baselines defined in experiments 1 and 2. This experiment also gives the best F-measure for subtask 2.

The rightmost column calculates the F-measure for both subtasks' F-measures. From these values we can conclude that using DicPRO together with contextual rules is a good choice for identifying anthroponyms, while giving a good precision in detecting proper names. Furthermore, the recall value obtained on the subtask 2 constitutes a base value, which may be subsequently improved with other methods. The introduction of statistical information of P-Dic made possible to obtain better results in subtask 2, but at the cost of impoverishing results of subtask 1, which was the main purpose for building DicPRO and the motivation for this experiment.

5 Conclusions and Future Work

Building lexical databases for proper names has not been the main approach in NLP to deal with this kind of linguistic elements. One of the reasons for this is the general assumption that the set of proper names is potentially infinite. This may not be exact for all classes of proper names, and most probably is not as far as anthroponyms (especially first names) are concerned.

This paper described a methodology employed in the building of an electronic dictionary of proper names (DicPRO). The paper evaluated the usefulness of this resource in a specific task: the capitalization of anthroponyms and proper names in general. The main purpose of this task was to improve the reading quality of the output of a speech recognition system, where every word, including proper names appears in lower case. Of course, the usefulness of this new tool could be extended to other scenarios where information regarding proper names is lacking or has been removed and needs to be restored (automatic spellchecking, named entities recognition, just to cite a few).

We compared and combined several approaches, namely, the use of a probabilistic dictionary (P-Dic) based on the use of upper case in a training corpus, and the use of contextual rules based on the information of DicPRO. Results consistently show improved results benefiting from the use of DicPRO.

Nevertheless, we expect to get better results applying automatic learning techniques like decision lists, such as described in [5,6], which are reported to achieve much better results for problems of similar nature. Hopefully, this will also help to enlarge the dictionary.

Furthermore, we intend to expand DicPRO, possibly by making use of extant lists of names and of lists of recognized named entities, the latter already made available for Portuguese in recent Named Entity Recognition (NER) evaluation contest, HAREM [5]. DicPRO should also evolve in order to encompass other types of proper names (toponyms, hidronyms and the like), and to integrate both simple and compound forms. By the expansion of the DicPRO coverage, we expect to apply this tool to the NER task in the near future.

References

1. Fourour, N., Morin, E, Daille, B.: Incremental recognition and referential categorization of French proper names. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), vol. III, pp. 1068-1074, (2002)
2. 2.Traboulsi, H.: A Local Grammar for Proper Names. MPhil Thesis. Surrey University (2004)
3. McDonald, D. Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In: Boguraev, B., Putejovsky, J. (eds.): Corpus Processing for Lexical Acquisition. pp. 61-76. MIT Press, Cambridge, Mass.(1993)
4. Friburger, N., Maurel, D.: Finite-state transducer cascades to extract named entities in texts. Theoretical Computer Science, Volume 313(1): 93-104 (2004)
5. Yarowsky, D.: Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. Proceedings of ACL'94, pp. 88-95 (1994).

[5] http://www.linguateca.pt/HAREM/

6. Yarowsky, D. Hierarchical Decision Lists for Word Sense Disambiguation. Computers and the Humanities, 34(1-2): 179-186. 2000
7. Piton, O., Maurel, D. : Les noms propres géographiques et le dictionnaire PROLINTEX. C. Muller, J. Royauté e M. Silberztein (eds.) INTEX Pour la linguistique et le traitement automatique des langues. Cahiers MSH Ledoux 1, pp. 53-76. Presses Universitaires de Franche-Comté: Besançon (2004).
8. Moura, P.: Dicionário electrónico de siglas e acrónimos. MSc Thesis, Faculdade de Letras da Universidade de Lisboa (2000, unpublished).
9. D. Caseiro, I. Trancoso: "Using dynamic wfst composition for recognizing broadcast news", in proc. ICSLP '2002, Denver, Colorado, EUA (2002).
10. Marie-Noël Gary-Prieur (ed.) Syntaxe et sémantique des noms propres. Langue Française 92. Larousse : Paris (data).
11. S. Leroy. Le nom propre en français. Paris: Ophrys (2004).
12. Jean Molino (ed.) Le nom propre. Langue Française 66.: Paris: Larousse (data)
13. Anderson, J.: On the Grammar of names. (to appear in Language 2004/05)
14. Silberztein, M. Dictionnaires électroniques et analyse automatique de texts. Le système INTEX. Masson, Paris (1993)
15. Trancoso, I.: "The ONOMASTICA Inter-Language Pronunciation Lexicon" Proceedings of EUROSPEECH'95 - 4th European Conference on Speech Communication and Technology - Madrid, Spain, September 1995.
16. Ranchhod, E., Mota, C., Baptista, J.: A Computational Lexicon of Portuguese for Automatic Text Parsing. SIGLEX-99: Standardizing Lexical Resources, pp. 74-80. ACL/Maryland Univ., Maryland (1999)
17. Baptista, J.: A Local Grammar of Proper Nouns. Seminários de Linguística 2: pp. 21-37. Faro: Universidade do Algarve (1998).

REPENTINO – A Wide-Scope Gazetteer for Entity Recognition in Portuguese

Luís Sarmento, Ana Sofia Pinto, and Luís Cabral

Faculdade de Engenharia da Universidade do Porto (NIAD&R),
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
las@fe.up.pt
Linguateca – Pólo do Porto, Portugal Via Panorâmica s/n, 4150-564 Porto
asofia@letras.up.pt
Linguateca – Pólo de Oslo,
P.O. BOX 124, Blindern, 0314 Oslo, Norway
Luis.M.Cabral@sintef.no

Abstract. In this paper we describe REPENTINO, a publicly available gazetteer intended to help the development of named entity recognition systems for Portuguese. REPENTINO wishes to minimize the problems developers face due to the limited availability of this type of lexical-semantic resources for Portuguese. The data stored in REPENTINO was mostly extracted from corpora and from the web using simple semi-automated methods. Currently, REPENTINO stores nearly 450k instances of named entities divided in more than 100 categories and subcategories covering a much wider set of domains than those usually included in traditional gazetteers. We will present some figures regarding the current content of the gazetteer and describe future work regarding the evaluation of this resource and its enrichment with additional information.

1 Introduction

The importance of Named Entity Recognition (NER) systems has been growing with the widespread of information extraction systems and applications. The goal of NER is to identify and correctly classify all Named Entities that exist in a given text according to a given predefined hierarchy or ontology. Broadly speaking, Named Entities (NE) include all entities that may be identified by a proper name, such as, for example, people, organizations, places, brands or products and other more abstract classes such as knowledge domains, techniques, or intellectual products (e.g.: "Computational Linguistics" or "9th Symphony"). The classification of numeric and time references is also usually included in the NER task. However, the detail and complexity of this task has varied greatly and has evolved over time. For example, in the first NER evaluation programs during MUC-6 [1], systems were asked to identify and classify entities belonging to a small set of generic categories, namely Person, Organization or Location. More recent evaluation programs, such as the ACE [2] or the Portuguese evaluation effort HAREM [3], required systems to perform classification over more detailed two-level hierarchies, to determine the semantic role of the referenced entities and to deal with other complex contextual constructions [4].

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 31 – 40, 2006.

Most NER systems are built using two possible strategies: (i) gazetteers and a set of manually encoded rules or (ii) by inferring classification rules from previously annotated corpora using supervised machine learning (ML) methods. In both cases important language resources are required (i.e. gazetteers or annotated corpora). Unfortunately, some languages lack publicly available resources for these purposes and adapting existing resources from other languages may require an effort equivalent to that of building the resource from scratch. Portuguese is one of those languages where the lack of resources has been problematic for the development of NER systems. Therefore, developing such resources should be considered a strategic option for the research community working with the computational processing of Portuguese. Additionally, since the definition of the NER task is rapidly expanding to include many more categories than the traditional ones (organization, location, person and numeric expressions), existing resources, when available, may not be enough to cover these latest requirements, which demand wider-scope gazetteers.

In this paper we will present REPENTINO (REPositório para reconhecimento de ENTIdades com NOme), a new publicly available gazetteer we have been developing that is organized according to a wide hierarchy that includes 11 top categories and 97 subcategories. Currently, REPENTINO stores more than 450000 instances of NE that have been extracted mainly from a large document collection and also from several thematic Web sites. REPENTINO has been manually validated to ensure the quality of the resource and is now freely available online in XML format from http://www.linguateca.pt/repentino/.

2 Motivation

Our motivation for building REPENTINO came from the difficulties encountered during the development of our own NER system: we were not able to find appropriate gazetteers for our NER system, either because they did not cover all the categories we were considering or because, for some categories, they were not comprehensive enough, covering only a small fraction of the cases. We thus began studying the possibility of building our own gazetteer, by making use of simple extraction techniques. The kind of techniques we were considering consisted of searching large quantities of text for archetypical lexical patterns that could lead us to instances of named-entities. For example, the lexical pattern “located in [Uppercased string]” could be used to identify instances of geographical entities. Although this approach seems quite naïve at first, simple tests allowed us to confirm that it is actually possible to extract hundreds of instances of organizations, locations and events from corpora with such simple techniques. Most importantly, instances could be validated without too much manual effort. Such procedures have, of course, their own limitations: it is very difficult to extract instances of some classes, such as product brands (e.g.: “luxury yachts”) or companies, because the contexts in which they appear are more diverse and more difficult to identify. But on the other hand, there are innumerous web sites where it is possible to find long lists of instances of such NE, and in some cases it is quite easy to harvest them manually by simple "copy-and-paste" methods. These two possibilities seemed promising enough to invest some effort in building a wide scope database of manually classified NE instances, for NER purposes.

3 Related Work

The need for wide scope classification systems capable of dealing simultaneously with various Information Extraction scenarios has been pointed out by Sekine [5]. The authors present an extended NE classification hierarchy, which includes 150 different types of NE organized in a 3 level tree structure, aiming to be comprehensive enough to cover major newspaper domains. This hierarchy is intended to classify a wide range of possible NE, including even some entities that may occur without capitalization (e.g. "leukemia"). In a later work [6], this hierarchy was extended to 200 categories. The authors also developed a dictionary containing 130000 instances of named-entities, organized according to the hierarchy developed. Instances were manually compiled from the Web, newspaper and other sources. The hierarchy was populated considering only the surface form of the entities. For example, "Portugal" would be considered a Place, although it may adopt different senses depending on the context (e.g.: "Organization"). In order to deal with several frequent cases of ambiguity in NE classification (e.g.: museums, theatres as either "Places" or "Organizations"), the hierarchy has several diffuse categories intended to classify such ambiguous instances.

Other recent works focus on dynamically building open classification hierarchies. Pasca [7] describes a system that is capable of finding both the NE instances and the corresponding (multiple) categories, using a lightly supervised Machine Learning (ML) algorithm. The author argues that traditional approaches to NE classification face the strong limitation of using closed categorization hierarchies, which most of the times are too coarse for dealing with flexible information extraction scenarios, namely web search. In those cases, categories are very diverse, overlapping, and very specific, which makes the process of developing a pre-defined category and the corresponding gazetteer unfeasible. Starting from a set of domain independent extraction patterns, the system is able to find categorized instances of named entities, and to obtain new contextual extraction patterns to find more categorized instances. The system is able to infer both general and very specific categories (and to obtain the corresponding instances) such as "colors", "hybrid cars" or "operating systems".

Our work for developing the classification system of REPENTINO and acquiring the corresponding NE instances lies somewhere between the top-down strategy followed by Sekine and the bottom-up approach of Pasca's work. Because of this, during the development of REPENTINO's hierarchy, we faced similar problems to those described in [5] such as for example deciding if a given instance should imply the creation of a new class in the system or could it be easily fit in an existing one. At the same time, our strategy for compiling instances of NE to populate REPENTINO has some points in common with the techniques described in [7] - although using manual processes instead of ML techniques – and has lead us to include several categories in the classification structure that we would have never otherwise predicted.

4 Structuring REPENTINO

The most complex questions when developing lexical-semantic resources are related to data organization. In developing a wide scope, fine-grained resource those questions involve dealing with philosophical assumptions about how the world should be organized. We knew of very detailed NE hierarchies, like [5] and [6], but such fine-grained hierarchies are very rare and, to our knowledge, there are no generic guidelines available for building them. There are many difficult questions related to developing adequate classification structures. For instance, in hierarchical structures, deciding if a given category should be split in several more specific may not be trivial and usually leads to some arbitrary decision. One should also note that, because of the ambiguous nature of many entities, a simple taxonomic hierarchy may not be an adequate structure at all, and may lead to difficult decisions regarding the classification of certain instances that could easily be placed in more than one category. Multiple inheritance connections may help to solve some of these questions but this usually leads to more complex classification systems. In fact, the whole issue of developing a classification structure is even more complex than that since any classification structure implies a specific ontology. However, any ontology (when agreed upon) is usually application-dependent, so committing to a given ontology may reduce the generality and portability of the resource.

Therefore, for building REPENTINO we followed three basic assumptions. The first assumption is that the classification structure should reflect the instances actually found in corpora or on the web, and should not be a pre-defined closed hierarchy, which usually involves making several ontological commitments. We thus decided not to adopt a pre-defined closed hierarchy but, instead, to follow an ad-hoc strategy for expanding an open set of categories, based on the instances that we were able to actually collect from corpora and from the Web by the processes described in the next sections. The structure of REPENTINO may be seen as a “loose” two-level hierarchy, with several generic top categories where more specialized sub-categories are spawned as new interesting NE instances are found. The hierarchy is not based on any strong ontological relations between top-level categories and their sub-categories. We tried to remove as many ontological assumptions as possible from the classification structure to make REPENTINO’s content reusable in several scenarios, and to circumvent hard philosophical questions. Sub-categories are considered the core of REPENTINO: instances are directly connected to subcategories and top-level categories which exist mainly for convenience reasons. The sub-categories could exist independently of the top-level categories, for example, as several separate specialized gazetteers. Ontological relations among instances or sub-classes are outside the scope of REPENTINO, and, if needed, they should be implemented by a particular application.

The second assumption is that instances found would always be classified according to their surface structure, i.e. considering the most immediate sense of the instance, and totally disregarding the several possible senses in context. Ambiguous cases, such as the place / organization ambiguity, once decided for a particular instance (e.g.: “Teatro Nacional D. Maria II” as a “cultural place” and not as an “organization”), would automatically imply that all similar cases would be classified equally (e.g. “Teatro de São João” would also be classified as a “cultural place”). For example, countries are stored in REPENTINO as a specific type of place. Ontological inferences, such as “a country may be seen as an organization in certain contexts”, are not in any way implied in REPENTINO, and depend solely on the application that uses the information stored in REPENTINO.

The third assumption is that REPENTINO stores instances of names rather than instances of individual entities or facts. This is indirectly related to how homograph instances should be dealt with. For example “America” may refer (at least) to a continent or to Franz Kafka’s book. Obviously, these two instances should be stored separately in REPENTINO, under the two different corresponding subcategories (in this cases Location-Terrestrial and Art/Media/Communication-Book as it will become clear later). But let us assume that there is another book named “America”. Should we store a second entry “America” under the same subcategory we place Kafka’s book before? The answer is negative because REPENTINO is intended to store names, not facts. REPENTINO should simply provide the information that there is (at least) one book named “America” (or that “America” could refer to an existing book) but not that “America” is a book by Franz Kafka, or by any other author.

5 Building REPENTINO

The actual process developing REPENTINO was very dynamic and was guided by particular problems faced during the development of our NER system. Whenever a given entity could not be correctly classified by our NER system - for example a luxury yacht brand - rather than trying to create a rule to deal with this case, we would search corpora or the Web for more instances of similar entities. This allowed us to obtain a broader picture of the problem and good insights about whether those instances should be added to REPENTINO or not.

This strategy affected dramatically the development of REPENTINO’s hierarchy. For instance, we were thus lead to create 16 subcategories under the category Location, almost all of which with more than 100 instances, and some with more than a thousand instances. But more importantly, we were able to include some very frequently mentioned named entities - such as Infrastructure/Facility or Real-Estate - that are rarely considered in most NE classification hierarchies. Similar situations happened for other top categories. It was also possible to compile many other instances that allowed us to include in REPENTINO totally unorthodox categories. For instance, REPENTINO includes a top category named “Paperwork” which we were able to fill with about 4500 instances, divided into eight subcategories.

5.1 Collecting NE Using Simple Patterns and Corpora

For extracting instances of NE from free text, we used BACO [8], a text database generated from the 14Gb WPT03 collection (http://www.linguateca.pt/wpt03/). The WPT03 collection is quite recent so it is very appropriate for extracting instances of relevant NE. However, for the extraction process to be feasible, we needed to be able not only to identify lists of possible NE instances, but also to have very strong clues about their categories to reduce the effort of manual validation. We thus tried to explore morphological and contextual properties of the (Portuguese) NE:

1. a typical head-word. Most of the entities have typical head-words, i.e. the first words of the NE are very typical and can almost certainly define its category: “Universidade do Porto”, or “Junta de Freguesia de Ramalde”.
2. an archetype context or collocation. There are many archetype contexts or collocations that may be used to extract certain types of NE. For example, in looking for locations, we may try to find what matches certain patterns such as “localizado na XXX” (“located in XXX”) or “próximo da XXX” (“near XXX”), where XXX has a very high probability of being a location.
3. a typical end-word. Some entities, such as companies and other organizations, may have certain typical end-words or acronyms. For example, in Portuguese, company names frequently end with particles such as “Lda.” or “S.A”.

Searches were performed using simple Perl scripts. Each complete run took approximately two hours, including manual revision, and we were usually able to extract up to 1000 instances of NE per run (sometimes many more).

5.2 Retrieving Instances from the Web

For some specific NE categories we found that it was much easier to find domain specific sites and collect some of the published information. For example, there are many sites on the web containing huge lists of movies, music and software applications. Such information is very difficult to extract from corpora, especially because it is not frequent enough, but it is readily available in some web sites. We were able to retrieve information from over 120 websites, taking advantage of several thematic ones, which did not have to necessarily be Portuguese. For example, names of software products, movie stars from the sixties or of luxury yachts can be compiled from sites in many possible languages. Apart from large scope sites such as the Portuguese and English version of Wikipedia, a great deal of our collecting effort was done over domain specific sites as, for example, sites from stock exchange markets. The choice of these domain specific sites was done in a rather ad hoc way. Some of them were found after searching the web using a regular search engine with a set of seed entities or by explicitly entering search expressions such as “list of celebrities”.

Other resourceful sites were found by trying links from well-known institutional sites. For example, we were able to find lists of several pharmaceutical products and active chemical substances visiting the web site of the national pharmaceutical administration office. Despite the apparent randomness of the process that led to many dead ends, this strategy proved to be an appropriate technique for collecting instances of NE that could not be easily retrieved from corpora. We believe that this allowed us to greatly improve the diversity of REPENTINO.

6 The “Loose” Classification Hierarchy of REPENTINO

Presently, the REPENTINO hierarchy comprises 11 top categories and 97 subcategories. Note that many of the subcategories are not likely to be considered when building a hierarchy using a top-down approach. However, by the processes explained before, we were able to retrieve large quantities of instances for such categories, which justifies their inclusion in REPENTINO. We will now present the current categories and subcategories and provide a brief explanation about them.

Location
Entities that are individualized essentially according to their position in the Universe. This category comprises the following subcategories: Terrestrial, Hydrographic, Address, Loose Address, Country/State, Town/Region/Administrative Division, Space, Socio-Cultural, Religious, Civil/Administration/Military, Heritage/Monuments, Other, Real-Estate, Mythological/Fictional, Commercial/Industrial/Financial, Infrastructure/ Facility.

Organizations
Entities that are composed by more than one person and that exist and operate as a whole. Organizations usually have goals, a set of rules and an internal structure that rule them, as opposed to simple groups of people or gatherings. Organizations are divided in the following subcategories: Company, Government/Administration, Education/R&D, Sports, Socio-Cultural, Interest Groups, Religious, Civil/Military, Clubs, Section, Other.

Beings
Real or fictional beings, as well as myths and mythological beings. Additionally, groups of people that do not explicitly form an Organization, such as ethnic and geopolitical groups, are also part of this category. In this hierarchy, the difference between Fictional beings and Myths is mainly that Fictional characters have never existed while Myths are not guaranteed to have existed or not. Also, a separate subcategory is considered for mythological beings, which are not the same as Myths. Beings are divided in the following subcategories: Human, Human-Collective, Non-Human, Geopolitical/Ethnic/Ideological, Mythological, Other.

Event
Events whose beginning and time span are clearly defined. Events include the following subcategories: Ephemerid, Cyclic, Scientific, Socio-Cultural, Sports, Political, Prize/Award, Other.

Products
This category includes many possible entities, ranging from industrial products to handcrafted objects. Note that although products and organizations may have a very similar name, there is an important difference between a Product and an Organization, since a Product should refer to a specific model, while organization is its producer. Products can be divided in the following subcategories: Brands, Consumables, Electronics/Appliances, Financial, Format, Gastronomic, Inspection/Exam, Services and Resources, Computer Systems and Applications, Clothing/Utilities, Vehicles, Medical/Pharmaceutical, Tools/Instruments, Craftwork, Other.

Art/Media/Communication
This is a specialized category that deals uniquely with products related to art, media and communication. Art/Media/Communication comprises the following subcategories: Books, Movies, TV/Radio/Theatre, Music, Fine-Arts & Design, Multimedia, Periodical, Scientific/Academic Paper, Other.

Paperwork
Laws, Decrees, Treaties, Pacts, Standards, Rules, Documents, Taxes and alike should be included in this category. This category can be divided in eight subcategories:

Laws, Certificates, Documents, Taxes/Fees, Proof/Test/Evaluation, Agreements, Standards, Other.

Substance
In this category we include elements, substances and minerals. Substances can be divided in the following subcategories: Group, Ore, Substance, Other.

Abstraction
Abstract entities such as disciplines, sciences, crafts, as well as certain mental formulations. We also include specific time periods, movements, states, diseases and crimes. Abstractions can be divided into the following subcategories: Disciplines/ Crafts, Period/Movement/Trend, State or Condition, Mental Formulation, Symbols, Crime, Latin Expressions, Era, Process, Type/Class, Index/Tax, Other.

Nature
This category includes animals, vegetables, all the elements that constitute living beings, as well as natural phenomena. Nature can be divided in five subcategories: Animal, Physiology, Micro-organisms, Vegetable, and Natural Phenomena.

Miscellanea
In this category we include words or symbols that are susceptible to collocate or to be present in the near context of some of the previous entities such as personal titles, currency and units.

7 Current Figures Behind REPENTINO

REPENTINO stores nearly 450000 instances of NE (a complete and updated statistical analysis is available on the web site). Currently, around 288K of the instances stored in REPENTINO (about 65%) belong to the category Beings. The distribution of the remaining instances is given in the next chart:

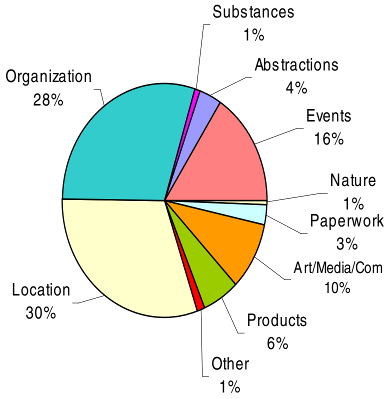


Fig. 1. – The distribution of instances according to the top categories

Apart from the category Beings, most of the instances stored in REPENTINO are Locations, Events, and Organizations, which seem to occur very frequently in the

WPT03 document collection. Other categories, such as Products are more difficult to obtain because they do not follow a strict morphology and, therefore, could not be so easily found by pattern matching processes.

8 Evaluation and Future Work

We have not yet performed any specific evaluation of REPENTINO, so no direct evaluation data is available at this moment. Direct evaluation of REPENTINO seems rather difficult because the value of this resource should be seen in relation to the success in Information Extraction tasks for which it was originally developed. At this level, some good indications about REPENTINO may be obtained by examining the results of the SIEMÊS [9], our NER system, in the recent HAREM evaluation contest. SIEMÊS heavily relied on REPENTINO as its main gazetteer and since it was one of the top scoring systems we may assume that some of its success is due to REPENTINO. A more direct evaluation of REPENTINO would have to focus on measuring specific values, such as for example the amount of overlap between its content and a gold standard, a corpus or other similar gazetteers. This will be object of future work. Other future improvements in REPENTINO aim at expanding the information stored in REPENTINO for NER purposes. For example by using a large document collection, or the Web, we may obtain information about the number of occurrences of each instance in REPENTINO and to retrieve corresponding contexts that may be used for developing rules in future NER classification procedures. Additionally, and following some of the ideas reported in [10], it seems useful to obtain information about which instances co-occur and from there try to determine possible NE clusters. Such information could be helpful for implementing new NE disambiguation procedures.

9 Conclusions

In this paper we have presented REPENTINO, a novel publicly available resource that may help researchers in the development of NER systems for Portuguese. REPENTINO was built using simple and semi-automatic NE extraction methods over large document collections, and also by manually searching the web. REPENTINO stores approximately 450000 manually validated instances of NE, organized in a loose two-level hierarchy with 11 top categories and 97 subcategories. REPENTINO has already been used in a practical NER system, whose performance was tested in the recent HAREM evaluation contest with positive results, so we believe it can be of great interest to the community developing technology for Portuguese.

Acknowledgements

This work was partially supported by grants POSI/PLP/43931/2001 and SFRH/BD/23590/2005 from Fundação para a Ciência e Tecnologia (Portugal), co-financed by POSI. The authors wish to thank Débora Oliveira for her help in the compilation process.

References

1. Grishman, R., Sundheim, B.: Message Understanding Conference - 6: A Brief History In Proc. Int. Conf. on Computational Linguistics, Copenhagen (1996) pp. 466-471.
2. Doddington, G., Mitchell A., Przybocki, M., Ramshaw, l., Strassel, S, Weischedel, R.: The Automatic Content Extraction (ACE) Program: Tasks, Data, and Evaluation. In: Proc. 4th Int. Conf. on Language Resources and Evaluation, Lisboa (2004) pp. 837-840.
3. Santos D., Seco N., Cardoso N., Vilela R.: HAREM: An Advanced NER Evaluation Contest for Portuguese. In: Proc. 5th Int. Conf. on Language Resources and Evaluation, Genoa, Italy (2006).
4. NIST. 2004. EDT Guidelines for English V4.2.6. http://www.ldc.upenn.edu/Projects/ACE/docs/EnglishEDTV4-2-6.PDF
5. Sekine, S., Sudo K., Nobata, C.: Extended Named Entity Hierarchy. In: Proc. 3rd Int. Conf. on Language Resources and Evaluation, Las Palmas, Canary Islands, Spain (2002).
6. Sekine, S., Nobata C.: Definition, dictionaries and tagger for Extended Named Entity Hierarchy. In: Proc. 4th Int. Conf. on Language Resources and Evaluation, Lisboa, Portugal, (2004) pp. 1977-1980.
7. Pasca, M.: Acquisition of categorized named entities for web search. In: Proc. 13th ACM Conf. on Information & Knowledge management. Washington, D.C., USA (2004) 137-145.
8. Sarmento, L.: BACO – A large database of text and co-occurrences. In: Proc. 5th Int. Conf, on Language Resources and Evaluation, Genoa, Italy (2006).
9. Sarmento, L: SIEMÊS – a Named-Entity Recognizer for Portuguese Relying on Similarity Rules. In: Proc. PROPOR 2006 - Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada. Itatiaia, RJ Brasil (2006).
10. Hasegawa, T, Sekine S., Grishman R.: Discovering Relations among Named Entities from Large Corpora. In: Proc. Annual Meeting of Association of Computational Linguistics (ACL 04). Barcelona, Spain (2004) pp. 415-422.

Using Natural Alignment to Extract Translation Equivalents[*]

Pablo Gamallo Otero

Departamento de Língua Espanhola, Faculdade de Filologia,
Universidade de Santiago de Compostela, Galiza, Spain
pablogam@usc.es

Abstract. Most methods to extract bilingual lexicons from parallel corpora learn word correspondences using relative small aligned segments, called sentences. Then, they need to get a corpus aligned at the sentence level. Such an alignment can require further manual corrections if the parallel corpus contains insertions, deletions, or fuzzy sentence boundaries. This paper shows that it is possible to extract bilingual lexicons without aligning parallel texts at the sentence level. We describe a method to learn word translations from a very roughly aligned corpus, namely a corpus with quite long segments separated by "natural boundaries". The results obtained using this method are very close to those obtained using sentence alignment. Some experiments were performed on English-Portuguese and English-Spanish parallel texts.

1 Introduction

To automatically extract a bilingual lexicon from a parallel corpus, we need first to align it, i.e., the various pieces or segments of text must be put into correspondence. This makes the lexicon extraction easier and more reliable. Many statistical algorithms for extraction of bilingual lexicon entries use statistical information derived from a sentence-aligned clean parallel corpus [5, 9, 1, 15, 12, 16, 8] The main drawback of this algorithm is that they are not very reliable when corpora have unclear sentence boundaries or the bilingual texts contain deletions, insertions and merged paragraphs. As [4] reported, many bilingual corpora contain noisy translations. In such a case, the aligned texts need to be manually corrected. It means that most approaches to bilingual dictionaries construction rely, to a certain extent, on supervised strategies.

The objective of this paper is to describe a specific method to compile bilingual lexicon entries from parallel corpora. Unlike most approaches cited above, we do not align texts at the sentence level. Bilingual texts are aligned using basic boundaries such as those separating chapters or articles. We call this strategy "natural alignment". As the detection of these long and *natural* segments is a trivial task that does not need further corrections, the extraction method we will describe is entirely unsupervised.

[*] This work has been supported by Ministerio de Educación y Ciencia of Spain, within the projects CESAR+ and GaricoTerm, ref: BFF2003-02866.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 41–49, 2006.

To achieve the same accuracy as comparable approaches to bilingual lexicon extraction, the correlation metric we propose in this paper will be well-suited to account for longer aligned segments than sentences. The correlation metrics used by most related work measure, given two words, if they co-occur in the list of paired sentences. These metrics rely on a boolean test: they count the times two words do and do not co-appear in the correlated sentences. Such a test is not suitable for long segments, where several tokens of the same word can occur. Our metric, on the contrary, takes into account the fact that a word can occur several times in a single (but long) segment.

Using this metric in several experiments, we verified that the accuracy of the extraction method remains stable when the parallel corpus contains a reduced number (no more than $2,000-3,000$) of aligned segments. The fact of increasing the number of segments does not improve accuracy in a significant way.

The main advantages of the method proposed in this paper are the following: First, lexicon extraction can be done from noisy bilingual corpus without making any correction (it is an entirely unsupervised method). Second, since we need very few (but long) aligned segments, the system turns out to be computationally more efficient than the standard strategies relying on many aligned sentences. And third, accuracy is not lower than in comparable approaches. In sum, a bilingual lexicon acquired using our method is, at least, as complete and accurate as a lexicon extracted from a sentence-aligned corpus. So, it can be used as a set of reliable anchor points to incrementally search for smaller aligned segments, until fine-grained word or multi-word level is achieved.

The remaining paper is organised as follows. In Section 2, we discuss some related work. Then, Section 3 introduces the notion of natural alignment. Section 4 defines the similarity coefficient proposed, and finally, in Section 5, some relevant experiments are performed and evaluated. The experiments were performed on both English-Spanish and English-Portuguese corpora.

2 Related Work

Most work on bilingual lexicon acquisition primarily relies on text aligned at the sentence level. The first approaches for sentence alignment used length-based algorithms [2, 3]. These algorithms assume that both the source and target sentences in the parallel corpus will be of constantly related length. Whereas [2] defined sentence length taking into account the number of words within a sentence, [3] defined length by counting the number of characters. Sentence delimiters are full stops. These strategies perform well only if the parallel texts have clear sentence boundaries. But full stops do not always mark sentence boundaries. Problems also arise when the target text contains insertions or deletions.

Other approaches to sentence-alignment use text based algorithms. They try to exploit more types of *correspondence points*, also called *anchor points*. The most common anchor points used by these approaches are the sequences of characters with the same form in both texts (e.g., homographs such as proper names and numbers) [12], or with similar forms (e.g., cognates as “president” and “presidente" in English and Spanish) [11, 13, 10]. Other anchor points can be taken from bilingual dictionaries or even from previously known translation equivalents. Anchor points are more robust to insertion and deletion than length-based algorithms. However, the main problem of text-based approaches is the large number of noisy points that make alignment less precise. To find reliable correspondence points, it is necessary to define heuristics to filter out those points that are not correct anchors.

Instead of elaborating complex strategies to filter out noisy correspondence points, we consider natural boundaries as reliable delimiters for parallel text alignment.

3 Natural Alignment

We align the source and target texts by detecting natural boundaries such as chapters of novels, directives of laws, articles of journals, articles of constitutions, etc. As natural boundaries preserve the basic structure of the document, they are more reliable anchor points than punctuation marks such as full stops or paragraph delimiters. For instance, the English translation of "El Quijote", found in the free catalogue of the Gutenberg project[1], contains 1,639 paragraphs whereas the source text is up 4,000. This makes automatic sentence alignment difficult even for well-known software aligners such as *Vanilla*[2]. By contrast, the two parallel texts - i.e., the source novel and its English translation - share the same number of chapters. So, we do not need to make further manual corrections if the alignment is made at the level of chapters.

The main drawback of this type of alignment is obviously the fact that it often selects long segments. Two advantages, however, deserve to be mentioned. First, deletions and additions found in some parts of the source texts do not prevent the correct alignment of the whole parallel corpus. And second, this alignment does not need manual correction. To overcome the problems derived from an alignment containing long segments, we posit a particular version of the Dice correlation metric.

4 Correlation Metric

To extract bilingual dictionaries from aligned texts, we need to measure the co-occurrence of words in the different aligned segments.

Following [14], we consider the Dice coefficient as an appropriate metric to measure translation correlations between word types. Given a source word w_1 and a candidate translation w_2, our particular version of the Dice coefficient is defined as follows:

$$Dice(w_1, w_2) = \frac{2F(w_1, w_2)}{F(w_1) + F(w_2)}$$

[1] http://www.gutenberg.org/

[2] http://nl.ijs.si/telri/Vanilla/

where

$$F(w_1, w_2) = \sum_i min(f(w_1, s_i), f(w_2, s_i))$$

and

$$F(w_n) = \sum_i f(w_n, s_i)$$

Note that $f(w_1, s_i)$ represents the frequency of the word type w_1 occurring in segment s_i. Unlike most approaches to bilingual lexicon extraction, we consider that the frequency of a word in a particular segment carries a very significant information. As the segments we use to align the corpus are longer as those used for sentence alignment, then, the same word can occur several times in the same segment. So, we assume a word of the target language, w_2, is likely to be a translation of a source expression, w_1, if they tend to have a similar frequency in each segment s_i. This is the main difference with regard to standard approaches. In most approaches, as segments are supposed to be small, the metric only takes into account boolean searches. More precisely, they check if the source word and its candidate translation do or do not appear in the same aligned segments. However, as in our approach many different words can occur at least once in all segments, we need a more informative feature, namely the number of times a word occur in each segment.

Table 1. Word vectors with frequency information

have	51	11	16	7	...
haber	46	8	13	4	...
estar	1	2	3	1	...

Table 2. Word vectors with boolean information

have	1	1	1	1	...
haber	1	1	1	1	...
estar	1	1	1	1	...

Let's take an example. Table 1 shows the first 4 positions of three word vectors: "have", "haber", and "estar". They were defined using the *EuroParl* parallel corpus (English-Spanish).

Concerning the verb "have", the first aligned segment contains 51 verb occurrences, the second segment 11, and so on. As natural segments are quite large, we need to take into account the number of occurrences in each segment in order to learn that "haber", and not "estar", is the likely translation of "have". Our particular definition of $F(w_1, w_2)$ allows us to grasp such a behaviour. By contrast, if the word vectors were defined using only a boolean test ("a word do and do not appear in a segment"), it would be impossible to distinguish between "haber" and "estar", since both Spanish words would be 100% similar to "have". See Table 2.

5 Experiments and Evaluation

Our objective is to test whether natural segments are useful to extract correct translation equivalents. For this purpose, we need to observe the behaviour of both the length and the number of aligned segments in the extraction process. Two different experiments were performed.

5.1 First Experiment

Table 3 shows the results of our first experiment. We used two different English-Spanish parallel corpora: the European Parliament proceedings, *EuroParl* [7], and the European Constitution. We selected a bitext of 1 million word tokens from EuroParl and then performed two different alignments. On the one hand, we used as natural segments the speaker interventions in the European Parliament. As a result, we obtained 3,000 aligned segments, where the average length of each segment is 2 KBytes. On the other hand, we aligned the same corpus using shorter segments (paragraphs). The number of paragraphs does not differ in the two languages because they were previously filtered by Koehn [7]. As paragraphs are 4 times shorter than the speaker interventions, this alignment gave rise to 12,000 segments. Concerning the European Constitution, we also aligned the corpus in two different ways. First, the alignment was performed using as natural segments the articles themselves. As a result, we obtained 881 segments. Second, we made an alignment at the sentence level by using the *Vanilla* aligner. We obtained 6,357 sentences. Each sentence is about 8 times shorter than an article.

For each aligned corpus, we generated a bilingual lexicon of single words tagged as nouns. The English corpus was tagged using TreeTagger[3], whereas we used FreeLing[4] to tag the Spanish text. The strategy to select the best translation equivalent was the following. For an English word, we selected the Spanish word with the highest Dice coefficient if and only if this coefficient is not lower than an empirically set threshold (0.3). Notice that, in this work, as we are not interested in evaluating the extraction strategy itself but only the input data (many and short against few and long segments), we used a very simple and intuitive extraction algorithm. In [6], we defined a more complex and efficient method to extract translation equivalents which takes into account word polysemia.

To evaluate the results, *precision* is characterised as the ability of the system to assign the correct translation equivalent to a word type in the dictionary. Let's note that the evaluation is made at the word type level. So, this *precision* does not consider the fact that a word can have several senses in different contexts. A translation proposed by the system is considered as correct if only if it is the correct translation in at least one particular context. *Recall* is the number of equivalents proposed by the system divided by the total number of word types found in the text. *F-score* is the harmonic mean between recall and precision.

[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

[4] http://garraf.epsevg.upc.es/freeling/

Table 3. Evaluation of two parallel corpus aligned in two different ways

Partitions	Segments	Precision	Recall	F-Score
EuroParl (speakers)	3,000	.77	.30	.43
EuroParl (paragraphs)	12,000	.85	.23	.32
Constit. (articles)	881	.69	.33	.45
Constit. (sentences)	6,357	.76	.29	.42

Table 3 shows that the fact of increasing the number of segments (by making their length shorter) has two effects: on the one hand, it improves precision, but, on the other, it makes recall lower. Notice that the f-score values obtained with the longest segments (speakers in EuroParl and articles in the Constitution) are higher than those obtained with small segments (paragraphs and sentences). These results seem to show that the mean between precision and recall (the f-score) is more balanced when the lexicon extraction is performed using fewer segments with a larger size, as in natural alignment.

5.2 Second Experiment

To check this idea, we made a more reliable experiment. We selected an excerpt from the English-Portuguese parallel corpus built within the Translation Project[5]. In order to tag the Portuguese corpus, we parametrised TreeTagger for Portuguese using the training corpus built by the NILC group ("Núcleo Interinstitucional de Linguística Computacional). As lexical resource, we made use of the "Léxico Multifuncional Computorizado do Português Contemporáneo[6]. The parametrised file for Portuguese is freely available[7].

The English-Portuguese parallel corpus we selected consists of 165,000 English word tokens and 28,000 manually aligned sentences. This corpus was splitted in 5 incremental partitions, where partition n is a subset of partition $n + 1$. The smallest partition represents only the 7% of the whole corpus; it consists of 2,000 segments. The second one is larger (18%) and consists of 5,000 segments. The third partition (36%) contains 10,000 segments. The forth (71%) has 20,000 segments, and finally, the firth partition is the whole corpus, that is, 28,000 aligned sentences. Table 4 depicts the precision, recall, and f-score achieved on these 5 partitions.

Later, this corpus was also aligned in 4,000 segments, where each segment is the result of putting 7 sentences together. In this experiment, the specific goal was to build an aligned corpus with few and long segments, as those usually generated by natural alignment. This aligned corpus was also splitted in 5 incremental partitions. The smallest one consists of 500 segments (12% of the

[5] This project aims at translating the Linux manuals to several languages. See more details in http://www.iro.umontreal.ca/translation/HTML/index.html

[6] The trained corpus is available at `http://www.nilc.icmc.usp.br/nilc/tools/nilctaggers.html`. The lexicon can be downloaded at `http://www.clul.ul.pt/sectores/projecto_lmcpc.html`.

[7] http://gramatica.usc.es/∼gamallo/tagger.htm.

Table 4. Evaluation of the 5 partitions extracted from the parallel corpus built by the Translation Project. Each aligned segment consists of 1 sentence.

Partitions	Segments	Precision	Recall	F-Score
1 (7%)	2,000	.84	.17	.28
2 (18%)	5,000	.87	.21	.34
3 (36%)	10,000	.90	.23	.37
4 (71%)	20,000	.91	.19	.31
5 (100%)	28,000	.89	.19	.31

Table 5. Evaluation of the 5 partitions extracted from the parallel corpus built by the Translation Project. Each aligned segment consists of 7 aggregated sentences.

Partitions	Segments	Precision	Recall	F-Score
1 (12%)	500	.83	.20	.32
2 (25%)	1,000	.83	.22	.35
3 (50%)	2,000	.89	.24	.38
4 (75%)	3,000	.89	.19	.31
5 (100%)	4,000	.88	.20	.33

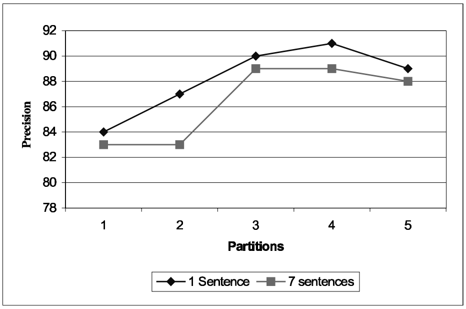


Fig. 1. Variation of precision as a function of corpus size

corpus). The second contains 1,000 (25%). The third and the forth consists of 2,000 and 3,000, respectively (50% and 75%). Finally, the largest partition embraces all the 4,000 segments. Table 5 shows the evaluation scores obtained on these partitions.

As in the experiments performed on both Europarl and the European Constitution, the alignment based on long segments entails a lower precision (.88 against .89 in the whole corpus), whereas recall is slightly better: .20 against .19. It results in a higher f-score: .33 against .31. These results confirm that the use of longer aligned segments provides a more balanced relationship between precision and recall than that produced by the alignment at the sentence level.

On the other hand, these experiments also show that precision becomes stable when the corpus partition reaches the 50% of the whole corpus (see Figure 1).

If we use sentence alignment, precision stabilises when the partition contains $10,000 - 20,000$ aligned segments. Yet, if alignment is made with longer portion of texts (as in natural alignment), then stability is achieved using less segments: $2,000$. In both experiments, given a specific corpus size, the fact of increasing the number of segments does not improve precision in a significant way.

However, these experiments are not enough to discover other relevant information. In particular, we need to perform further experiments to find the relationship between the size of aligned segments and the number of segments that are required to precision be stabilised.

6 Conclusions and Further Research

The experiments described above lead us to conclude that it is not necessary to align parallel corpora at the sentence level if the aim is only to extract bilingual translation equivalents. For this specific purpose, what we need is to identify natural boundaries and use the particular version of the Dice coefficient defined above. This type of strategy is robust to noise (insertions and deletions) in the parallel corpus and produces results that are at least as useful as those obtained by sentence alignment. Moreover, the bilingual lexicon extracted using our method could be considered as a set of reliable anchor points if the aim is to incrementally search for smaller aligned segments (until reach the word level).

Taking into account these conclusions and the evaluation results, nothing prevents us from building a huge amount of parallel texts whose basic unit of alignment is a book or a novel. For instance, consider that we are provided with a huge electronic library containing thousands of old Spanish novels and their English translations[8]. Then, a computationally realistic task will be to use our strategy to extract a bilingual lexicon taking as basic segment each single novel and its translation. Since such an aligned corpus is considerably bigger than those corpora aligned at the sentence level, the bilingual lexicon extracted from this enormous parallel corpus could reach an interesting coverage. As soon as big digital libraries are available, it will be possible to perform this type of experiment.

References

1. Lars Ahrenberg, Mikael Andersson, and Magnus Merkel. A simple hybrid aligner for generating lexical correspondences in parallel texts. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 29–35, Montreal, 1998.
2. P.F. Brown, J. Lai, and R. Mercer. Aligning sentences in parallel corpora. In *29th Conference of ACL*, 1991.
3. K. Church. Char_align: A program for aligning parallel texts at the character level. In *31st Conference of the Association for Computational Linguistics (ACL)*, pages 1–8, Columbus, Ohio, 1993.

[8] Some promising projects, such as the Google Library and the Gutenberg Project, are working to collect electronic books without Copyright in free libraries.

4. Pascale Fung and Kathleen McKeown. Finding terminology translation frmo non-parallel corpora. In *5th Annual Workshop on Very Large Corpora*, pages 192–202, Hong Kong, 1997.
5. Willian Gale and Kenneth Church. Identifying word correspondences in parallel texts. In *Workshop DARPA SNL*, 1991.
6. Pablo Gamallo. Extraction of translation equivalents from parallel corpora using sense-sensitive contexts. In *10th Conference of the European Association on Machine Translation (EAMT'05)*, pages 97–102, Budapest, Hungary, 2005.
7. Philipp Koehn. Europarl: A multilingual corpus for evaluation of machine translation. In *http://people.csail.mit.edu/koehn/publications/europarl/*, 2003.
8. Oi Yee Kwong, Benjamin K. Tsou, and Tom B. Lai. Alignment and extraction of bilingual legal terminology from context profiles. *Terminology*, 10(1):81–99, 2004.
9. Dan Melamed. A word-to-word model of translational equivalence. In *35th Conference of the Association of Computational Linguistics (ACL'97)*, Madrid, Spain, 1997.
10. Dan Melamed. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1), 1999.
11. A. Ribeiro, G. Dias, G. Lopes, and J. Mexia. Cognates alignment. In *Machine Translation Summit VIII*, pages 287–293, Santiago de Compostela, Spain, 2001.
12. A. Ribeiro, G. Lopes, and J. Mexia. Using confidence bands for parallel texts alignment. In *38th Conference of the Association for Computational Linguistics (ACL)*, pages 432–439, 2000.
13. M. Simard and P. Plamondon. Bilingual sentence alignment: Balancing robustness and accuracy. *Machine Translation*, 13(1):59–80, 1998.
14. F. Smadja, K. McKeown, and V. Hatzivassiloglou. Translating collocations for bilingual lexicons. *Computational Linguistics*, 22(1), 1996.
15. Jorg Tiedemann. Extraction of translation equivalents from parallel corpora. In *11th Nordic Conference of Computational Linguistics*, Copenhagen, Denmark, 1998.
16. Ŝ. Vintar. Using parallel corpora for translation-oriented term extraction. *Babel Journal*, 47(2):121–132, 2001.

Open-Source Portuguese–Spanish Machine Translation

Carme Armentano-Oller, Rafael C. Carrasco, Antonio M. Corbí-Bellot,
Mikel L. Forcada*, Mireia Ginestí-Rosell, Sergio Ortiz-Rojas,
Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez,
Felipe Sánchez-Martínez, and Miriam A. Scalco

Transducens Group,
Departament de Llenguatges i Sistemes Informàtics,
Universitat d'Alacant, E-03071 Alacant, Spain
*`mlf@ua.es`

Abstract. This paper describes the current status of development of an open-source shallow-transfer machine translation (MT) system for the [European] Portuguese ↔ Spanish language pair, developed using the OpenTrad Apertium MT toolbox (`www.apertium.org`). Apertium uses finite-state transducers for lexical processing, hidden Markov models for part-of-speech tagging, and finite-state-based chunking for structural transfer, and is based on a simple rationale: to produce fast, reasonably intelligible and easily correctable translations between related languages, it suffices to use a MT strategy which uses shallow parsing techniques to refine *word-for-word* MT. This paper briefly describes the MT engine, the formats it uses for linguistic data, and the compilers that convert these data into an efficient format used by the engine, and then goes on to describe in more detail the pilot Portuguese↔Spanish linguistic data.

1 Introduction

This paper presents the current status of development of an open-source (OS) shallow-transfer machine translation (MT) system for the [European] Portuguese ↔ Spanish language pair, developed using the recently released OpenTrad Apertium MT toolbox (`http://www.apertium.org`), Apertium for short. Apertium is based on an intuitive approach: to produce fast, reasonably intelligible and easily correctable translations between related languages, it suffices to use a MT strategy which uses shallow parsing techniques to refine fixed-equivalent, *word-for-word* machine translation.

Apertium uses finite-state transducers for lexical processing (powerful enough to treat many kinds of multi-word expressions), hidden Markov models (HMM) for part-of-speech tagging (solving categorial lexical ambiguity), and finite-state-based chunking for structural transfer (local *structural* processing based on simple and well-formulated rules for some simple structural transformations such as word reordering, number and gender agreement, etc.).

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 50–59, 2006.

This design of Apertium is largely based on that of existing systems such as interNOSTRUM[1] (Spanish↔Catalan, [1]) and Tradutor Universia[2] (Spanish↔Brazilian Portuguese, [2]), systems that are publicly accessible through the net and used on a daily basis by thousands of users.

The Apertium toolbox has been released as OS software;[3] this means that anyone having the necessary computational and linguistic skills can adapt it to a new purpose or use it to produce a MT system for a new pair of related languages.

In addition to the toolbox, OS data are available for three language pairs: Spanish–Catalan and Spanish–Galician, developed inside the OpenTrad consortium,[4] and more recently, Spanish–European Portuguese, developed by the authors and described in this paper. Prototypes for all three pairs may also be tested on plain, RTF, and HTML texts and websites at the address `http://www.apertium.org`.

The introduction of open-source MT systems like Apertium may be expected to ease some of the problems of closed-source commercial MT systems: having different technologies for different pairs, closed-source architectures being hard to adapt to new uses, etc. It will also help shift the current business model from a licence-centered one to a services-centered one, and favor the interchange of existing linguistic data through the use of standard formats.

The Spanish↔Portuguese language pair is one of the largest related-language pairs in the world; this is one of the main reasons to release pilot OS data for this pair. We believe this may motivate researchers and groups to improve these data or adapt them to other variants of Portuguese such as Brazilian Portuguese, and collaborate to develop, in the near future, a high-quality, free, OS Portuguese↔Spanish MT system.

This paper briefly describes the MT engine (sec. 2), the formats it uses for linguistic data (sec. 3), the compilers that convert these data into an efficient format used by the engine (sec. 4), and the pilot Spanish↔Portuguese linguistic data (sec. 5). Brief concluding remarks are given in sec. 6.

2 The Apertium Architecture

The MT strategy used in the system has already been described in detail [1, 2]; a sketch (largely based on that of [3]) is given here.

The MT engine is a classical shallow-transfer or *transformer* system consisting of an 8-module assembly line (see figure 1); we have found that this strategy is sufficient to achieve a reasonable translation quality between related languages such as Spanish and Portuguese. While, for these languages, a rudimentary word-for-word MT model may give an adequate translation for 75% of the text,[5]

[1] `http://www.internostrum.com`

[2] `http://traductor.universia.net`

[3] Under the GNU General Public License, `http://www.gnu.org/licenses/gpl.html`

[4] `http://www.opentrad.org`

[5] Measured as the percentage of the words in a text that do not need correction

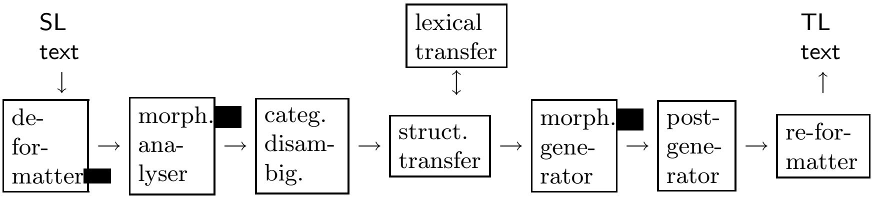


Fig. 1. The eight modules of the Apertium MT system (see section 2)

the addition of homograph disambiguation, management of contiguous multiword units, and local reordering and agreement rules may raise the fraction of adequately translated text above 90%. This is the approach used in the engine presented here.

To ease diagnosis and independent testing, modules communicate between them using text streams (see examples below). Most of the modules are capable of processing tens of thousands of words per second on current desktop workstations; only the structural transfer module lags behind at several thousands of words per second. A description of each module follows.

The de-formatter: separates the text to be translated from the format information (RTF and HTML tags, whitespace, etc.). Format information is encapsulated in brackets so that the rest of the modules treat it as blanks between words. For example, the HTML text in Portuguese "`vi <em>a bola</em>`" ("I saw the ball") would be transformed by the de-formatter into "`vi[ <em>]a bola[</em>]`".[6]

The morphological analyser: tokenizes the text in *surface forms* (lexical units as they appear in texts) and delivers, for each surface form (SF), one or more *lexical forms* (LFs) consisting of *lemma*, *lexical category* and *morphological inflection information*. For example, upon receiving the example text in the previous section, the morphological analyser would deliver

```
^vi/ver<vblex><ifi><1><sg>$[ <em>]
^a/a<pr>/o<det><def><f><sg>/o<prn><pro><3><f><sg>$
^bola/bola<n><f><sg>$[</em>]
```

where each SF has been analysed into one or more LFs: *vi* is analysed into lemma *ver*, lexical category *lexical verb* (`vblex`), indefinite indicative (`ifi`), 1st person, singular; *a* (a homograph) receives three analyses: *a*, preposition; *o*, determiner, definite, feminine singular ("the"), and *o*, proclitic pronoun, 3rd person, feminine, singular ("her"), and *bola* is analyzed into lemma *bola*, noun, feminine, singular. The characters "`^`" and "`$`" delimit the analyses for each SF; LFs for each SF are separated by "`/`"; angle brackets "`<`...`>`" are used to delimit grammatical symbols. The string after the "`^`" and before the first "`/`" is the SF as it appears in the source input text.[7]

[6] As usual, the escape symbol \ is used before symbols `[` and `]` if present in the text.

[7] The \ escape symbol is used before these special characters if present in the text.

Tokenization of text in SFs is not straightforward due to the existence, on the one hand, of contractions, and, on the other hand, of multi-word lexical units. For contractions, the system reads in a single SF and delivers the corresponding sequence of LFs (for instance, the Portuguese preposition-article contraction *das* would be analysed into two LFs, one for the preposition *de* and another one for the article *as*). Multi-word SFs are analysed in a left-to-right, longest-match fashion; for instance, the analysis for the Spanish preposition *a* would not be delivered when the input text is *a través de* ("through"), which is a multi-word preposition in Spanish.

Multi-word SFs may be invariable (such as multi-word prepositions or conjunctions) or inflected (for example, in Portuguese, *tinham saudades*, "they missed", is a form of the imperfect indicative tense of the verb *ter saudades*, "to miss"). Limited support for some kinds of inflected discontinuous multiword units is also available. The module reads in a binary file compiled from a source-language (SL) morphological dictionary (see section 3).

The part-of-speech tagger: As has been shown in the previous example, some SFs (about 30% in Romance languages) are homographs, ambiguous forms for which the morphological analyser delivers more than one LF; when translating between related languages, choosing the wrong LF is one of the main sources of errors. The part-of-speech tagger tries to choose the right LF according to the possible LFs of neighboring words. The part-of-speech tagger reads in a file containing a first-order hidden Markov model (HMM, [4]) which has been trained on representative SL texts. Two training modes are possible: one can use either a larger amount (millions of words) of untagged text processed by the morphological analyser or a small amount of tagged text (tens of thousands of words) where a LF for each homograph has been manually selected. The second method usually leads to a slightly better performance (about 96% correct part-of-speech tags, considering homographs and non-homographs). The behavior of the part-of-speech tagger and the training program are both controlled by a tagger definition file (see section 3). The result of processing the example text delivered by the morphological analyser with the part-of-speech tagger would be:

```
^ver<vblex><ifi><1><sg>$[ <em>]
^a<det><def><f><sg>$
^bola<n><f><sg>$[</em>]
```

where the correct LF (determiner) has been selected for the word *a*.

The lexical transfer module: is called by the structural transfer module (described below); it reads each SL LF and delivers a corresponding target-language (TL) LF. The module reads in a binary file compiled from a bilingual dictionary (see section 3). The dictionary contains a single equivalent for each SL entry; that is, no word-sense disambiguation is performed. For some words, multi-word entries are used to safely select the correct equivalent in frequently-occurring fixed contexts.[8] This approach has been used with very good results in Tradutor Universia and interNOSTRUM. Each of the LFs in the running example would be translated into Spanish as follows:

```
ver<vblex> : ver<vblex>
o<det> : el<det>
bola<n><f> : balón<n><m>
```

where the remaining grammatical symbols for each LF would be simply copied to the TL output. Note the gender change when translating *bola* to *balón*.

The structural transfer module: uses finite-state pattern matching to detect (in the usual left-to-right, longest-match way) fixed-length patterns of LFs (chunks or phrases) needing special processing due to grammatical divergences between the two languages (gender and number changes to ensure agreement in the TL, word reorderings, lexical changes such as changes in prepositions, etc.) and performs the corresponding transformations. This module is compiled from a transfer rule file (see below). In the running example, a *determiner–noun* rule is used to change the gender of the determiner so that it agrees with the noun; the result is

```
^ver<vblex><ifi><1><sg>$
[ <em>]^o<det><def><m><sg>$
^balón<n><m><sg>$[</em>]
```

The morphological generator: delivers a surface (inflected) form for each TL LF. The module reads in a binary file compiled from a TL morphological dictionary (see section 3). The result for the running example would be `vi[ <em>]el balón[</em>]`.

The post-generator: performs orthographic operations such as contractions and apostrophations. The module reads in a binary file compiled from a rule file expressed as a dictionary (section 3). The post-generator is usually *dormant* (just copies the input to the output) until a special *alarm* symbol contained in some TL SFs *wakes it up* to perform a particular string transformation if necessary; then it goes *back to sleep.* For example, in Portuguese, clitic pronouns in contact may contract: *me* ("to me") and *o* ("it", "him") contract into *mo*, or prepositions such as *de* ("of") may contract with determiners like *aquele* ("that") to yield contractions such as *daquele.* To signal these changes, linguists prepend an *alarm* symbol to the TL SFs *me* and *de* in TL dictionaries and write post-generation rules to effect the changes described.

The re-formatter: restores the format information encapsulated by the deformatter into the translated text and removes the encapsulation sequences used to protect certain characters in the SL text. The result for the running example would be the correct Spanish translation of the HTML text: `vi <em>el balón</em>`.

[8] For example, the Portuguese word "bola" ("ball") would be translated as "balón", but as "pelota" when it is part of the multiword unit "bola de tenis".

3 Formats for Linguistic Data

An adequate documentation of the code and auxiliary files is crucial for the success of OS software. In the case of a MT system, this implies carefully defining a systematic format for each source of linguistic data used by the system.

Apertium uses XML[9]-based formats for linguistic data for interoperability; in particular, for easier parsing, transformation, and maintenance. The XML formats for each type of linguistic data are defined through conveniently-designed XML document-type definitions (DTDs) which may be found inside the `apertium` package (available through `http://www.apertium.org`). On the one hand, the success of the OS MT engine heavily depends on the acceptance of these formats by other groups;[10] acceptance may be eased by the use of an interoperable XML-based format which simplifies the transformation of data from and towards it. But, on the other hand, acceptance of the formats also depends on the success of the translation engine itself.

Dictionaries (lexical processing): Monolingual morphological dictionaries, bilingual dictionaries and post-generation dictionaries use a common format.

Morphological dictionaries establish the correspondences between SFs and LFs and contain (a) a definition of the alphabet (used by the tokenizer), (b) a section defining the grammatical symbols used in a particular application to specify LFs (symbols representing concepts such as noun, verb, plural, present, feminine, etc.), (c) a section defining paradigms (describing reusable groups of correspondences between parts of SFs and parts of LFs), and (d) one or more labelled dictionary sections containing lists of SF–LF correspondences for whole lexical units (including contiguous multi-word units). Paradigms may be used directly in the dictionary sections or to build larger paradigms (at the conceptual level, paradigms represent the regularities in the inflective system of the corresponding language).

Bilingual dictionaries have a similar structure but establish correspondences between SL LFs and TL LFs.

Finally, *post-generation dictionaries* are used to establish correspondences between input and output strings corresponding to the orthographic transformations to be performed by the post-generator on the TL SFs generated by the generator.

Tagger definition: SL LFs delivered by the morphological analyser are defined in terms of fine part-of-speech tags (for example, the Portuguese word *cantávamos* has lemma *cantar*, category *verb*, and the following inflection information: *indicative, imperfect, 1st person, plural*), which are necessary in some parts of the MT engine (structural transfer, morphological generation); however, for the purpose of efficient disambiguation, these fine part-of-speech tags may be manually grouped in coarser part-of-speech tags (such as "verb in personal form"). In the tagger definition file (a) coarser tags are defined in terms of fine

[9] `http://www.w3.org/XML/`

[10] This is indeed the mechanism by which *de facto* standards appear.

tags, both for single-word and for multi-word units, (b) constraints may defined to forbid or enforce certain sequences of part-of-speech tags, and (c) priority lists are used to decide which fine part-of-speech tag to pass on to the structural transfer module when the coarse part-of-speech tag contains more than one fine tag. The tagger definition file is used to define the behavior of the part-of-speech tagger both when it is being trained on a SL corpus and when it is running as part of the MT system.

Structural transfer: rule files contain pattern–action rules describing what has to be done for each pattern (much like in languages such as `perl` or `lex` [5]). Patterns are defined in terms of categories which are in turn defined (in the preamble) in terms of fine morphological tags and, optionally, lemmas for lexicalized rules. For example, a commonly used pattern, *determiner–noun*, has an associated action which sets the gender and number of the determiner to those of the noun to ensure gender and number agreement.

De-formatters and re-formatters: are generated also from *format management files*. These are not linguistic data but are considered in this section for convenience. Format management files for RTF (rich text format), HTML (hypertext markup language) and plain text are provided in package `apertium`. The corresponding compilers generate C++ de-formatters and re-formatters for each format using `lex` [5] as an intermediate format.

4 Compilers

The Apertium toolbox contains compilers to convert the linguistic data into the corresponding efficient form used by the modules of the engine. Two main compilers are used in this project: one for the four lexical processing modules of the system and another one for the structural transfer.

Lexical processing: The four lexical processing modules (morphological analyser, lexical transfer, morphological generator, post-generator) are implemented as a single program which reads binary files containing a compact and efficient representation of a class of finite-state transducers (letter transducers, [6]; in particular, augmented letter transducers [7]). The lexical processor compiler [8] is very fast (it takes seconds to compile the current dictionaries in the system) which makes linguistic data development easy: the effect on the whole system of changing a rule or a lexical item may be tested almost immediately.

Structural transfer: The current structural transfer compiler (version 0.9.1 of `apertium`) reads in a structural transfer rule file and generates a C++ structural transfer module using `lex` [5] as an intermediate step. This makes it mandatory to recompile the engine each time the structural transfer data change; we are currently working on a precompiled format for transfer rules which would be read in by a general structural transfer module.

5 Portuguese↔Spanish Data

Lexical data: Currently, the Portuguese morphological dictionary contains 9700 lemmas; the Spanish morphological dictionary, 9700 lemmas, and the Spanish–Portuguese bilingual dictionary, 9100 lemma–lemma correspondences.

Lexical disambiguation: The tagset used by the Portuguese (resp. Spanish) HMM [4] lexical disambiguator consists of 128 (resp. 78) coarse tags (80 —resp. 65— single-word and 48 —resp. 13— multi-word tags for contractions, etc.) grouping the 13,684 (resp. 2,512) fine tags (412 (resp. 374) single-word and 13,272 (resp. 2,143) multi-word tags) generated by the morphological analyser.[11] The number of parameters in the HMM is drastically reduced by grouping words in ambiguity classes [4] receiving the same set of part-of-speech tags: 459 (resp. 260) ambiguity classes result. In addition, a few words such as *a* (article or preposition) or *ter* (*to have*, auxiliary verb or lexical verb) are assigned special hidden states. The Spanish lexical disambiguator has similar figures.

The current Portuguese (resp. Spanish) disambiguator has been trained as follows: initial parameters are obtained in a supervised manner from a 29,214-word (resp. 22,491-word) hand-tagged text and the resulting tagger is retrained (using Baum-Welch re-estimation as in [4]) in an unsupervised manner over a 454,197-word (resp. 520,091-word) text. Using an independent 6,487-word (resp. 24,366-word) hand-tagged text, the observed coarse-tag error rate is 4,0% (resp. 2,9%).

Before training the tagger we forbid certain impossible bigrams, such as *ter* as a lexical verb (translated into Spanish as *tener*) before any participle, so that in that case, *ter* is translated as an auxiliary verb (translated as *haber*).

Structural transfer data: The Portuguese↔Spanish structural transfer uses about 90 rules (the Spanish↔Portuguese figures are similar). The main group of rules ensures gender and number agreement for about 20 very frequent noun phrases (determiner–noun, numeral–noun, determiner–noun–adjective, determiner–adjective–noun etc.), as in *um sinal vermelho* (Portuguese, masc.) [“a red signal”]) → *una señal roja* (Spanish, fem.). In addition, we have rules to treat very frequent Portuguese–Spanish transfer problems, such as these:

- Rules to ensure the agreement of adjectives in sentences with the verb *ser* (“to be”) to translate, for example, *O sinal é vermelho* (Portuguese, masculine, “The signal is red”) into *La señal es roja* (Spanish, feminine).
- Rules to choose verb tenses; for example, Portuguese uses the subjunctive future (*futuro do conjuntivo*) both for temporal and hypothetical conditional expressions (*quando vieres* [“when you come”], *se vieres* [“if you came”]) whereas Spanish uses the present subjunctive in temporal expressions (*cuando vengas*) but imperfect subjunctive for conditionals (*si vinieras*).
- Rules to rearrange clitic pronouns (when enclitic in Portuguese and proclitic in Spanish or vice versa): *enviou-me* (Portuguese) → *me envió* (Spanish)

[11] The number of fine tags in Portuguese is high due to mesoclitics in verbs.

["he/she/it sent me"]; *para te dizer* (Portuguese) → *para decirte* (Spanish) ["to tell you"], etc.
- Rules to add the preposition *a* in some modal constructions (*vai comprar* (Portuguese) → *va a comprar* (Spanish) ["is going to buy"]).
- Rules for comparatives, both to deal with word order (*mais dois carros* (Portuguese) → *dos coches más* (Spanish) ["two more cars"]) and to translate *do que* (Portuguese) ["than"] as *que* (Spanish) in Portuguese comparative constructs such as *mais... do que...*
- Lexical rules, for example, to decide the correct translation of the adverb *muito* (Portuguese) → *muy/mucho* (Spanish) ["very", "much"] or that of the adjective *primeiro* (Portuguese) → *primer/primero* (Spanish) ["first"].
- A rule to translate the progressive Portuguese structure *estar a* + infinitive into the Spanish structure *estar* + gerund (`en` *to be* + *-ing*), and vice versa.
- Rules to make some syntactic changes like those needed to correctly translate the Portuguese construction "gosto de cantar" ("I like to sing") into Spanish as "me gusta cantar". Note that simple syntactic changes can be performed despite Apertium does not perform syntactic analysis.

Post-generation data: Current post-generation files for Spanish contain 26 entries using 5 paradigms grouping common post-generation operations. The most common Spanish post-generation operations include preposition–determiner contractions, or using the correct form of Spanish coordinative conjunctions *y/e*, *o/u* depending on the following vowel. On the other hand, current post-generation files for Portuguese contain 54 entries with 16 paradigms grouping common post-generation operations. Portuguese post-generation operations include clitic–clitic, preposition–determiner, and preposition–pronoun contractions.

A quick evaluation: With the described data, the text coverage of the Portuguese–Spanish (resp. Spanish–Portuguese) system is 92.8% (resp. 94,3%) as measured on a 5,294-word (resp. 5,028-word) corpus gathered from various sources. The translated word error rate (including unknown words) is 10.5% (resp. 8.3%). Speed surpasses 5000 words per second on an desktop PC equipped with a Pentium IV 3 GHz processor.

6 Concluding Remarks

We have presented the application of the OpenTrad Apertium open-source shallow-transfer machine translation toolbox to the Portuguese–Spanish language pair. Promising results are obtained with the pilot open-source linguistic data released (less than 10000 lemmas and less than 100 shallow transfer rules) which may easily improve (down to error rates around 5%, and even lower for specialized texts), mainly through lexical contributions from the linguistic communities involved. Note that the OpenTrad Apertium open-source engine itself is still being actively developed and contributions to its design may enhance it to perform more advanced lexical and structural processing tasks so that it can deal with more general language pairs.

Acknowledgements. Work funded by the Spanish Ministry of Industry, Tourism and Commerce through grants FIT-340101-2004-0003 and FIT-340001-2005-2, and partially supported by the Spanish Comisión Ministerial de Ciencia y Tecnología through grant TIC2000-1599-CO2-02. Felipe Sánchez-Martínez is supported by the Ministry of Education and Science and the European Social Fund through graduate scholarship BES-2004-4711.

References

1. Canals-Marote, R., Esteve-Guillen, A., Garrido-Alenda, A., Guardiola-Savall, M., Iturraspe-Bellver, A., Montserrat-Buendia, S., Ortiz-Rojas, S., Pastor-Pina, H., Perez-Antón, P., Forcada, M.: The Spanish-Catalan machine translation system interNOSTRUM. In: Proceedings of MT Summit VIII: Machine Translation in the Information Age. (2001) Santiago de Compostela, Spain, 18–22 July 2001.
2. Garrido-Alenda, A., Gilabert Zarco, P., Pérez-Ortiz, J.A., Pertusa-Ibáñez, A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Scalco, M.A., Forcada, M.L.: Shallow parsing for Portuguese-Spanish machine translation. In Branco, A., Mendes, A., Ribeiro, R., eds.: Language technology for Portuguese: shallow processing tools and resources. Edições Colibri, Lisboa (2004) 135–144
3. Corbí-Bellot, A.M., Forcada, M.L., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Alegria, I., Mayor, A., Sarasola, K.: An open-source shallow-transfer machine translation engine for the romance languages of Spain. In: Proceedings of the Tenth Conference of the European Association for Machine Translation. (2005) 79–86
4. Cutting, D., Kupiec, J., Pedersen, J., Sibun, P.: A practical part-of-speech tagger. In: Third Conference on Applied Natural Language Processing. Association for Computational Linguistics. Proceedings of the Conference, Trento, Italy (1992) 133–140
5. Lesk, M.: Lex — a lexical analyzer generator. Technical Report 39, AT&T Bell Laboratories, Murray Hill, N.J. (1975)
6. Roche, E., Schabes, Y.: Introduction. In Roche, E., Schabes, Y., eds.: Finite-State Language Processing. MIT Press, Cambridge, Mass. (1997) 1–65
7. Garrido-Alenda, A., Forcada, M.L., Carrasco, R.C.: Incremental construction and maintenance of morphological analysers based on augmented letter transducers. In: Proceedings of TMI 2002 (Theoretical and Methodological Issues in Machine Translation, Keihanna/Kyoto, Japan, March 2002). (2002) 53–62
8. Ortiz-Rojas, S., Forcada, M.L., Ramírez-Sánchez, G.: Construcción y minimización eficiente de transductores de letras a partir de diccionarios con paradigmas. Procesamiento del Lenguaje Natural (35) (2005) 51–57

Weighted Finite-State Transducer Inference for Limited-Domain Speech-to-Speech Translation

Diamantino Caseiro and Isabel Trancoso

L^{2}F INESC-ID/IST
{Diamantino.Caseiro, Isabel.Trancoso}@inesc-id.pt
http://www.L2F.inesc-id.pt

Abstract. A speech input machine translation system based on weighted finite state transducers is presented. This system allows for a tight integration of the speech recognition with the machine translation modules. Transducer inference algorithms to automatically learn the translation module are also presented. Good experimental results confirmed the adequacy of these techniques to limited-domain tasks. In particular, the reordering algorithm proposed showed impressive improvements by reducing the error rate in excess of 50%.

1 Introduction

Weighted finite-state transducers (WFSTs) provide a very compelling framework for speech-to-speech translation, because of the successful use of WFSTs to integrate multiple sources of knowledge in automatic speech recognition. A large variety of knowledge sources, from common ones such as acoustic models, lexica and language models [1], to unusual ones such as pronunciation rules [2], have all been represented using this framework. Furthermore, the use of weighted transducer composition and other optimizing operations [1] allows the tight integration of the knowledge sources in a single optimized search space. By treating the translation module as an additional knowledge source, it too can be tightly integrated in a speech-input machine translation system.

Various machine translation systems based on finite-state transducers have also been proposed. For example, [3] shows how the IBM 3 [4] statistical translation model can be implemented using WFSTs. In [5] a composition-based approach is shown that builds a translation system by representing it as the composition of a lexical translation with a reordering transducer. Our system is based on transducer inference techniques. These kind of techniques have also been used successfully by various authors [6, 7, 8].

In the next section the architecture of the proposed speech-to-speech translation system is presented, followed in section 3, by the description of the translation algorithms used. The Portuguese/English parallel corpus used to evaluate the system is presented in section 4, while the experimental results are shown in section 5. The main conclusions are presented in section 6.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 60–68, 2006.

2 System Architecture

Our automatic speech recognition system [9] imposes very few constraints on its search space. The main requirement is that it must consist of a single transducer mapping from acoustic tags (usually corresponding to distributions of acoustic features) to output language words. However, in practice this search space is built as the composition of multiple WFSTs representing knowledge sources such as acoustic models ($\mathcal{H}$), lexical models ($\mathcal{W}$), language models ($\mathcal{G}$) or translation models ($\mathcal{T}$). The systems makes few assumptions regarding the topology of the search space, and other knowledge sources, such as context dependency, and pronunciation rules can be easily integrated as long as they are implemented as WFSTs.

One additional advantage of the use of WFSTs is that the search space can be replaced by an equivalent but more efficient transducer. This optimization is performed offline through the use of operations such as weighted determinization, minimization and pushing, see [1].

The size of models in real speech-to-speech tasks is so great that their integration in the search space, as described above, would make their use unpractical. Due to this reason, our decoder uses specialized algorithms so that the generation and optimization of the search space is done *on-the-fly* [10]. The combined transducer is expanded on demand while processing the input signal so that only the necessary parts are explicitly represented. The computational overhead of these mechanisms is modest, and they allow a reduction of the memory needs to a practical level.

Our decoder is thus very flexible, since different model can be combined freely in a composition cascade. The simplest configuration is to use a composition of the acoustic, lexical and translation models: $\mathcal{H} \circ \mathcal{W} \circ \mathcal{G}$. Here the translation model is acting also as a language model for the input and output language. By integrating the translation transducer in the cascade, the input language speech signal is recognized directly into output language words.

3 Machine Translation Algorithms

In start this section by presenting the general transducer inference framework used by out machine translation algorithms. Then the algorithms themselves will be presented.

3.1 Transducer Inference

In general inference of languages is easier than inference of translations. The literature about inference of transducers or relations is small when compared with the many useful algorithms for learning finite-state automata or equivalent models.

One framework for developing transducer inference algorithms is the Grammatical Inference and Alignments for Transducer Inference (GIATI) family of algorithms [8].

Given a parallel corpus consisting of a finite set A of string pairs $(\overline{x}, \overline{y}) \in \Sigma^* \times \Delta^*$, a transducer inference algorithm can be obtained by following the 3 steps:

1. Each pair $(\overline{x}, \overline{y})$ from the training set A is transformed into a string $\overline{z}$ from an *extended alphabet* Γ yielding a sample S of strings, $S \subset \Gamma^*$. Each element of Γ is usually a pair of strings, so that this step effectively transforms a pair of strings into a string of pairs.
2. A grammar inference algorithm is applied to the sample S. Typically this results in a (stochastic) finite-state automaton $\mathcal{A}$.
3. Each symbol in $\mathcal{A}$ is split into an input and an output string, thus transforming it into a transducer $\mathcal{T}$.

The first step is modeled by a labeling function $\mathcal{L} : \Sigma^\star \times \Delta^\star \to \Gamma^\star$, while the last one is defined by an "inverse labeling function" $\Lambda(\cdot)$, such that $\Lambda(\mathcal{L}(A)) = A$.

$$\begin{tikzcd}
\begin{array}{c} A \subset \Sigma^\star \times \Delta^\star \\ \text{Training pairs} \end{array} \arrow[r, "\text{Step 1} - \mathcal{L}(\cdot)"] & \begin{array}{c} S \subset \Gamma^\star \\ \text{Training strings} \end{array} \arrow[d, "\text{Step 2}"] \\
\begin{array}{c} \mathcal{T}\colon A \subset T(\mathcal{T}) \\ \text{A finite-state transducer} \end{array} & \begin{array}{c} \mathcal{A}\colon S \subset L(\mathcal{A}) \\ \text{A finite-state automaton} \end{array} \arrow[l, "\text{Step 3} - \Lambda(\cdot)"']
\end{tikzcd}$$

Fig. 1. Basic scheme for the inference of finite-state transducers. A is a finite sample of training pairs. S is a finite sample of strings. $\mathcal{A}$ is an automaton inferred from S such that S is a subset of the language $L(G)$. $\mathcal{T}$ is a finite-state transducer whose translation $T(\mathcal{T})$ includes the training sample A.

Using this general framework, different algorithms can be instantiated by using different labeling, $\mathcal{L}$, techniques, or by using different grammar inference algorithms in the second step.

3.2 Step 1 - Labeling

Labeling is the main challenge of this approach. In general, many possible labeling transformations are possible, however, the existence of complex alignments among words makes the design of the transformation a difficult problem. On the other hand, this transformation must be reversible and allow for the implementation of a inverse labeling for the third step. An interesting way to address these problems is using the statistical translation framework techniques to align parallel corpora [4].

To illustrate the labeling process we will use the pair of sentences shown in figure 2.

INPUT: `por favor , tem algum quarto duplo livre ?`
OUTPUT: `do you have a double room available , please ?`

Fig. 2. Example sentence

INPUT: `NULL ( 2 ) por ( ) favor ( 9 ) , ( 8 ) tem ( 1 3 ) algum ( 4 ) quarto ( 6 ) duplo ( 5 ) livre ( 7 ) ? ( 10 )`
OUTPUT: `do you have a double room available , please ?`

Fig. 3. Example of a GIZA++ alignment

```
(por / ) (favor / ) (, / ) (tem / do you have) (algum / a) (quarto / )
(duplo / double room) (livre / available , please) (? / ?)
```

Fig. 4. Example of a string of pairs

Statistical alignments. The design of the labeling function $\mathcal{L}$ requires the ability to find relations between words in the input and output sentences. In general, we want each element of the extended alphabet Γ to consist of a pair of input and output strings that are translation equivalents. The relations can be captured by alignments. An alignment is a correspondence between words from the input text and words from the output text. The manual construction of alignments is an expensive and tedious task, and most often, algorithms are used to automatically build (statistical) alignments. Formally, an alignment between a pair of sentences $(\overline{x}, \overline{y})$ is a mapping $i \rightarrow j = a_i$ that assigns a word x_j in position j of the input string to a word y_i in position $i = a_j$ of the output string. Alignments are used as a hidden variable in statistical machine translation models such as IBM models [4], and can be inferred automatically using, for example, the expectation maximization (EM) algorithm.

In this paper, all alignments were obtained automatically using the IBM 4 model implemented in the GIZA++ [11] software package. Figure 3 shows an example of an alignment generated with this package. In the alignments, each input word is aligned with a set of output words, for example, "`quarto ({ 6 })`" means that input word "`quarto`" is associated with the sixth output word "`room`". To account for output words not associated with input words, a "`NULL`" word is added to each input sentence. In this example, no input word was associated with word "`you`".

Labeling. After generating alignments from the training data, it is necessary to convert them into corresponding strings of pairs, such as the one shown in figure 4.

The main idea is assigning each input word with the corresponding output words as given by the alignment. However, sometimes this assignment violates the word order of the output string. "`NULL`" words are also problematic. Techniques need to be developed to address these two problems.

The baseline algorithm was based on [12], and consists of the following 3 steps:

1. Merge output words aligned with ”NULL” with the previous output word, except words in position 1 which are merged with the word in position 2. After this step, the alignment in figure 3 is transformed into:

   ```
   NULL ({ }) por ({  }) favor ({ 9 }) , ({ 8 }) tem ({ 1 2 3 })
   algum ({ 4 }) quarto ({ 6 }) duplo ({ 5 }) livre ({ 7 })
   ? ({ 10 })}
   ```

2. Delay the production of out-of-order output words by associating them with the first input word where they can be produced in order. In the example, the association of output word in position 6 is moved from word ”quarto” to word ”duplo”, among other changes:

   ```
   NULL ({ }) por ({  }) favor ({ }) , ({ }) tem ({ 1 2 3 })
   algum ({ 4 }) quarto ({ }) duplo ({ 5 6 }) livre ({ 7 8 9 })
   ? ({ 10 })}
   ```

3. Convert the association of each word into a pair, and build the string of pairs:

   ```
   (por / ) (favor / ) (, / ) (tem / do you have) (algum / a)
   (quarto / )  (duplo / double room) (livre / available , please)
   (? / ?)
   ```

A different strategy can be used to deal with out-of-order words which consists in keeping the alignment word order, but adding additional information to the generated pairs to allow recovering the output language word order in a post-processing step. The idea is that when a out-of-order word is found, 3 relative offsets are considered: the offset relative to the current alignment order, the offset relative to the beginning of the output string and the offset relative to the end of the output string. The word is marked with the smallest of this 3 offsets. Using this strategy, we propose the following algorithm `reorder`:

1. Process ”NULL” words as in the baseline algorithm
2. Mark each out-of-order word with the offset of its correct order relative to the alignment order (or to the beginning or ending of the string):

   ```
   NULL ({ }) por ({  }) favor ({ -1E:9 }) , ({ -2E:8 })
   tem ({ 1 2 3 }) algum ({ 4 }) quarto ({ 1:6 }) duplo ({ -1:5 })
   livre ({ 7 }) ? ({ 10 })}
   ```

3. Convert the association of each word into a pair, as in the previous algorithm, thereby generating the following string of pairs:

   ```
   (por / ) (favor / -1E:please ) (, / -2E:,) (tem / do you have)
   (algum / a) (quarto / 1:room) (duplo / -1:double)
   (livre / available) (? / ?)
   ```

In this notation, `1:room` means that word `room` should be moved by one position to the right, while `-1E:favor` means that word `favor` should be moved to the penultimate position (-1 relative to the end `E` of the sentence). One should note that, during the grammar inference step of GIATI, pairs such as `(quarto / room)`, `(quarto / 1:room)` or `(quarto / 2:room)`, that differ only in the offsets, will be considered distinct.

3.3 Step 2 - Grammar Inference

The grammar inference algorithm used in our translation algorithms consists of smoothed n-grams models.

From the strings of pairs corpus an n-gram language model is estimated considering each pair as a token. The language model is then converted to a weighted finite-state automaton $\mathcal{A}$.

The conversion consists of encoding the context of each n-gram as a state, and encoding each n-gram as a transition between states. For example, $p(x_i|x_{i-1}x_{i-2})$ is encoded as a transition between states $(x_{i-1}x_{i-2})$ and (x_ix_{i-i}), with label x_i and weight $p(x_i|x_{i-1}x_{i-2})$. Smoothed n-gram models are encoded using ϵ transitions to encode backoffs [13].

All n-gram models used in this work were trained using the SRILM [14] software package.

3.4 Step 3 - Inverse Labeling

Inverse labeling is done by replacing the label of each arc in the automaton $\mathcal{A}$, by the two strings of the string pair encoded in the label.

4 Portuguese-to-English Corpus

The speech-to-speech task used here is inspired in the Spanish-to-English task developed in project EuTrans [15]. This is a tourist domain task and consists of typical human-to-human communication situation at the front desk of a hotel. The textual component of the corpus was generated in a semi-automatic way using travel booklets as a seed corpus. This resulted in a very large set of 490,000 sentence pairs, including many repetitions.

Our Portuguese-English corpus was generated by translating the Spanish portion of the semi-automatic generator used in the original Spanish-English version of EuTrans . The training corpus has 490,000 pairs, also with many repetitions of simple sentences: as in the original Spanish-English corpus, only approximately 170k are distinct. In order to approach more realistic conditions, a subset of the corpus was randomly selected, yielding a reduced text corpus of 10,000 sentences. All experiments reported in this paper were performed using this reduced subset.

Furthermore, in order to evaluate the system, a multi-speaker Portuguese speech corpus was collected for testing. No speech was collected for training purposes. The test set consists of 400 utterances: 100 from each of 4 different speakers (2 male and 2 female). The speech was collected using a head-mounted

Table 1. Tourist domain Portuguese-English Corpus

Data		Portuguese	English
Big training corpus	Sentence pairs	500,000	
	Different sentence pairs	169,847	
	Running words	4,463,000	4,850,000
	Vocabulary	939	772
	Bigram Test-Set Perplexity	8.2	5.8
Small training corpus	Sentence pairs	10,000	
	Different sentence pairs	6711	
	Running words	89445	93313
	Vocabulary	897	731
	Bigram Test-Set Perplexity	8.7	6.0
Test	Speech utterances	400	—
	Running words	3,700	—

microphone in a typical office environment. A summary of the main features of this Portuguese-English corpus is presented in Table 1.

5 Experiments

In the machine translation experiments reported in this paper, the translation of the input string was computed, and the translation word error-rate (TWER) was used as an error criterion. The TWER is a string edit distance computed as the minimum number of substitutions, deletions or insertions that have to be performed to convert the translated string into the (single) reference translation.

5.1 Machine Translation Results

In order to evaluate the quality of the translation models, they were tested with textual input. The translation of the input string was computed by converting it to a linear automaton $\mathcal{I}$ and computing $\mathit{bestpath}(\pi_2(\mathcal{I} \circ \mathcal{M}))$. That is, by composing $\mathcal{I}$ with a particular translation model $\mathcal{M}$, applying the output projection π_2 to convert the composition transducer to an automaton containing all possible translations in the model, and finally searching for the best one.

Table 2 summarizes the translation performance of various versions of the proposed algorithms. Translation transducers are designated by $\mathcal{T}$ when built using the baseline labeling algorithm and by $\dot{\mathcal{T}}$ when built using the reordering one; $\mathcal{G}$ is an output language models, and $\mathcal{R}$ is a reordering model[1]. The n-gram order of the various models is shown as a subscript.

The 3 first models in the table were build using the baseline algorithm. From these results we observe that increasing the order of the n-gram improves performance up to order 4, and that longer contexts result in worse results due to the relative small size of the training set.

[1] This model was not implemented as a transducer. Instead a small program was used to reorder the output strings.

Table 2. Text translation results

Model	TWER
$\mathcal{T}_3$	7.25 %
$\mathcal{T}_4$	6.19 %
$\mathcal{T}_5$	7.77 %
$\mathcal{T}_3 \circ \mathcal{G}_4$	5.96 %
$\mathcal{T}_4 \circ \mathcal{G}_4$	5.36 %
$\dot{\mathcal{T}}_3 \circ \mathcal{R}$	4.41 %
$\dot{\mathcal{T}}_4 \circ \mathcal{R}$	2.82 %

Table 3. Speech input results

Model	TWER
$\mathcal{H} \circ \mathcal{W} \circ \mathcal{T}_3$	13 %
$\mathcal{H} \circ \mathcal{W} \circ \mathcal{T}_4$	12 %
$\mathcal{H} \circ \mathcal{W} \circ \dot{\mathcal{T}}_4 \circ \mathcal{R}$	9 %

From the results of experiments $\mathcal{T}_3 \circ \mathcal{G}_4$ and $\mathcal{T}_4 \circ \mathcal{G}_4$, we see benefits from the use of an additional language model, although this configuration has no statistical justification.

The best results were obtained using the reorder algorithm, which had a dramatic impact in this, task reducing the TWER to half, relative to the baseline. One of the reasons for this improvements is the capability of this algorithm to account for long distance word reordering relations which are very frequent in this corpus. One example, is the relation between words "`por favor ,`" and "`, please`" in the example of figure 3.

5.2 Speech Input Machine Translation Results

The translation models created were also tested in a speech input task. The acoustic model used was not adapted to this task and was reused from a broadcast news task [16]. The pronunciation lexicon was mostly reused from that task, however 110 new pronunciation were added.

The speech recognition error rate (WER) of the system was 5.4% when configured using a Portuguese 3-gram language model ($\mathcal{H} \circ \mathcal{W} \circ \mathcal{G}_3$).

The translation performance varied between 13% and 9% TWER.

6 Conclusions

In this paper, a speech recognition system capable of combining finite-state models though composition was explored as the keystone of a speech-to-speech translation architecture. Translation models were developed based on grammatical inference and statistical techniques. In particular, the proposed reordering algorithm, although very simple, showed impressive improvements in this task, by reducing the translation error rate in excess of 50%.

Acknowledgments. We would like to thank Prof. Francisco Casacuberta for providing us the Spanish to English EUTRANS corpus and generator. This work was partially funded by FCT project POSI/PLP/47175/2002.

References

1. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. In: ASR 2000 Workshop, Paris, France (2000)
2. Hetherington, I.: An efficient implementation of phonological rules using finite-state transducers. In: Proc. Eurospeech '2001, Aalborg, Denmark (2001)
3. Knight, K., Al-Onaizan, Y.: Translation with finite-state devices. In: Lecture Notes in Computer Science 1529. Springer Verlag (1998)
4. Gale, W., Church, K.: A program for aligning sentences in bilingual corpora. Computational Linguistics **102** (1993) 19–75
5. Bangalore, S., Riccardia, G.: Stochastic finite-state models for spoken language machine translation. In: Workshop on Embedded Machine Translation Systems, Seattle, EUA (2000)
6. Oncina, J., Castellanos, A., Vidal, E., Jimenez, V.: Corpus–based machine translation through subsequential transducers. In: Third International Conference on the Cognitive Science of Natural Language Processing, Dublin, Ireland (1994)
7. García-Varea, I., Sanchis, A., Casacuberta, F.: A new approach to speech-input statistical translation. In: 15th International Conference on Pattern Recognition. Volume 3., Barcelona, Spain, IEEE Computer Society (2000) 94–97
8. Casacuberta, F., Vidal, E., Picó, D.: Inference of finite-state transducers from regular languages. Pattern Recognition **38** (2005) 1431–1442
9. Meinedo, H., Caseiro, D., Neto, J., Trancoso, I.: Audimus.media: a broadcast news speech recognition system for the european portuguese language. In: PROPOR'2003 - 6th International Workshop on Computational Processing of the Portuguese Language, Springer, Faro, Portugal (2003)
10. Caseiro, D.: Finite-State Methods in Automatic Speech Recognition. PhD thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa (2003)
11. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics **29** (2003) 19–51
12. Casacuberta, F.: Inference of finite-state transducers by using regular grammars and morphisms. In: Lecture Notes in Computer Science 1891. Springer Verlag (2000)
13. Riccardi, G., Bocchieri, E., Pieraccini, R.: Non deterministic stochastic language models for speech recognition. In: Proc. ICASSP '95, Detroit, USA (1995) 237 – 240
14. Stolcke, A.: Srilm - an extensible language modeling toolkit. In: Proc. ICSLP '2002, Denver, Colorado, USA (2002)
15. Amengual, J., Benedí, J., Casacuberta, F., Castaño, A., Castellanos, A., Jiménez, V., Llorens, D., Marzal, A., Pastor, M., Prat, F., Vidal, E., Vilar, J.: The eutrans-i speech translation system. Machine Translation **15** (2000) 75–103
16. Meinedo, H., Souto, N., Neto, J.: Speech recognition of broadcast news for the european portuguese language. In: Automatic Speech Recognition and Understanding ASRU'2001, Madona de Campilho, Trento, Italy (2001)

A Golden Resource for Named Entity Recognition in Portuguese

Diana Santos[1] and Nuno Cardoso[2]

[1] Linguateca: Node of Oslo at SINTEF ICT
[2] Linguateca: Node of XLDB at University of Lisbon

Abstract. This paper presents a collection of texts manually annotated with named entities in context, which was used for HAREM, the first evaluation contest for named entity recognizers for Portuguese. We discuss the options taken and the originality of our approach compared with previous evaluation initiatives in the area. We document the choice of categories, their quantitative weight in the overall collection and how we deal with vagueness and underspecification.

1 Introduction

HAREM [1] was the first evaluation contest for named entity recognition (NER) in Portuguese, gathering the community interested in named entity recognition, as well as developing a new methodology for the evaluation of NER systems. Following the well-known evaluation context paradigm [2, 3], participants cooperated in the design and population of the golden resource used in HAREM, the golden collection (GC), which we describe in detail here.

The golden collection is a set of texts with the correct identification and classification of the named entities according to the HAREM guidelines.[1] This resource – of which each participant had only seen a tiny extract – was then embedded in a much larger text collection, the HAREM collection, which was provided untagged to all participating systems, which in turn had two days to return it NE-annotated. Evaluation was performed by comparing results with the answers previously encoded in the golden collection.

Although this is a fairly standard model, already used for Portuguese in Morfolimpíadas [4, 5], there are some interesting ways in which this evaluation contest differs from MUC [6] and other predecessors in the NE domain:

Kind of text. HAREM encompassed a broad range of texts, of which language variety, genre and (implicit) publication mode was provided to the participants. In addition to allow a finer-grained characterization of the problem, such information allows more advanced systems to change behaviour according to any of these parameters.[2]

[1] Guidelines and collection are available from `http://www.linguateca.pt/HAREM/`.

[2] Although in HAREM's first edition we are not aware of any system profitting from this feature, in the future it is to be expected that textual genre is used by informed systems to select different interpretations of a particular NE.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 69–79, 2006.

Tailored evaluation. We allowed participants to choose to participate in all categories, or just to be evaluated for a subset of their own choice. We also gathered separate information for specific scenarios, per main category (e.g. if the system's task was only to find locations, or organizations in text) or comparing with MUC.

Bottom-up categories. Instead of using previous NE categories often designed for other languages, HAREM concerns a set of Portuguese linguistically motivated categories (and corresponding types), arrived at by careful consideration and manual annotation of a wide amount of text of different genres.

Context annotation. Emphasis was put on the context of use: HAREM was meant to evaluate the type of a named entity in the **context** it is used, not its general description type. For example, the name of a newspaper may be classified as an organization (company), a place (of publication) or a person (the interviewer) depending on the context. This clearly makes the task evaluated by HAREM more difficult,[3] but we believe such behaviour to be more useful in real applications.

Separation of identification and classification. We conceptually separated identification (e.g. the correct delimitation of the NE) from classification (which could be semantic or morphological).

Morphology. HAREM included a morphological task, since gender and number are properties of most Portuguese named entities, and may sometimes help semantic disambiguation.[4]

New measures. HAREM introduced several new measures to assess the performance of the systems, dealing with indeterminacy and with partially correctly identified NEs.

We start the paper by providing, in section 2, some background on the named entity task and our motivation for organizing HAREM. In section 3 we describe the golden collection in detail, touching upon identification guidelines, classification guidelines, dealing with indeterminacy, and problems we are aware of. In section 4, we provide some first analyses making use of this collection.

2 Context and Motivation

Named entity recognition was first coined in MUC 6, when it became an independent track in a larger effort to evaluate information extraction. Since then, several other evaluation contests have been organized around this task, for English as well as for other languages [9, 10]. More specific subsets of the original NE set, time expressions, have also been evaluated separately [11].

In fact, the organizers of evaluation campaigns like ACE [12] and dedicated workshops [13] have claimed that the time was ripe to investigate more challenging goals. However, although several groups had their in-house procedures and gazetteers, there was still no independent off-the-shelf NE component for Portuguese at the time work

[3] Other evaluation contests might consider "newspaper name" or "organization" as a correct label in every such case.

[4] [7] note that organizations named after male persons tend to retain the feminine gender, while football clubs are often described as masculine even if named after something originally feminine. [8] mention the acronym *EUA* as referring to the United States (masculine plural) or the European Universities Association (feminine singular).

was started in charting interest and tools in the area. See [7] for a detailed historical overview, providing also background and further motivation to identify and classify Portuguese NEs.

There were, anyway, other scientific and methodological reasons that led us to organize HAREM, in addition to the obvious relevance to the Portuguese language processing community:

First of all, there was not, as far as we know, any independent-from-English investigation of which needs (in terms of other categories) other languages might have in this respect. For example, would person, organization, and place names be all, only and forever the right categories to identify, as [9] seems to assume uncritically? Although Sekine et al. [14] have proposed an ontology large enough to encompass both English and Japanese, and others have looked into the problem of multilingual NER [15, 16], there was no bottom-up **monolingual** NER evaluation contest we knew of.[5] As shown in the remainder of this paper, we found a significant number of other relevant NEs to be catered for in Portuguese.

Second, would the NE categories used be equally relevant no matter the register or genre of texts, almost a decade later, if one wished to parse the Web, literary text and technical documents, too? We wanted to investigate whether the sort of NEs in a given text might be a discriminating feature for classifying genre [20]. Preliminary results on this subject have already been published in [1].

Finally, an issue in which we had a special interest is how much language-dependent NER is [21], and how much (if anything) can be gained by attacking Portuguese as opposed to language-general methods, which we expected international participants would try out. As to this last point, let us stress that HAREM is not comparing the same task for different languages, as [22, 23] have done. Rather, given a specific, particular, language (Portuguese), one of our goals was to measure, instead, how well language-dependent and language-independent methods fare.[6]

The reasons to organize HAREM were thus manifold, ranging from practical opportunity to theoretical relevance: As to the former, there was a relatively large number of interested parties, which is a requirement for a successful event. In addition, in contradistinction to morphological analysis [5] where it was hard to establish a common set of agreed categories, because many choices of the participating systems were only motivated by the different behaviours of the parsers using them, NER in itself boasted of a number of almost direct applications, although they differed significantly in their requirements.[7]

3 The Golden Collection

We defined the global task of HAREM as follows: systems had to classify all descriptions of entities involving capitalization in Portuguese, i.e. proper names, in addition to

[5] Although we were obviouly aware of a number of detailed proposals for a.o. German [17], Spanish [18] and Swedish [19].

[6] This analysis must be based on the methodologies employed by the different participants.

[7] The original task specification, as opposed to MUC, drastically differed among participants, interested e.g in deploying geographically-aware search engines, or searching and indexing oral documents or semi-automatically producing terminology.

expressions involving digits. So, capitalized section titles, textual emphasis, or common words that are spelled with capitals just because they originate from (other language's) initials, like *CD*, are **not** considered NEs. Expressions that should denote named entities but have been mispelled by not being capitalized are not considered NEs, either.

3.1 Purpose and Rationale

We attempted to include in the golden resource of HAREM the ideal set of marking needed by any text in Portuguese to represent the distinctions we were interested in. The annotation of the GC does not represent the desiderata an automatic system is supposed to achieve nowadays, or ever, nor is it assumed that most of the NEs should appear in a gazetteer – on the balance of several methods, see [24].

One of our goals with HAREM was, in fact, to measure the degree of difficulty of the general NER task in Portuguese, and not particularly the performance of real systems today. To arrive at a particular classification may require a huge amount of reasoning and world knowledge – probably something no present system is ready to do. Still, we wanted to estimate how often this is the case.

Finally, it is relevant to note that we strove to get at linguistically motivated categories in the GC. That is, we did not start from an ideal ontology on how the real world should be conceived, but we observed what kinds of references were typically capitalized in Portuguese texts. After a first mental organization of these different cases, we attempted to posit their specific subcategorization (what we call types) which objectively showed different linguistic properties in Portuguese. For example, reproducible works of art are usually encoded within quotes, while unique ones do not; atomic events are often compound names as in *Benfica-Sporting*, while organized ones receive a more “ordinary" proper name; and references to publications, or laws, appear usually in a special and very specific format in running text.[8]

The release (and use) of the GC will help us assess whether we succeeded in the goal of proposing only linguistically different types, or whether still many iterations are required to obtain further consensus and more replicable guidelines.

3.2 Quantitative Snapshot

HAREM's golden collection is a collection of texts from several genres (table 1) and origins (table 2), in which NEs have been identified, semantically classified and morphologically tagged in context, and revised independently, according to a large set of directives approved (and discussed) by all participants.

3.3 The Classification Grid

Our study of Portuguese texts uncovered that, in addition to locations (LOCAL), people (PESSOA) and organizations (ORGANIZACAO), as well as values (VALOR) and dates

[8] They may pose interesting linguistic problems as well. Susana Afonso (p.c.) noted that a reference with multiple authors behaves differently as far as morphology – and therefore type of reference – is concerned. E.g., we may say *Santos et al. (2006) mostraram* (plural animated subject, “S et al showed”) or *Cardoso and Santos mostraram em Cardoso & Santos (2006)* (singular inanimate location, “C&S showed in C&S”).

Table 1. Genre variety distribution

Genre	Text extracts	Words	NEs
Web	40 (30.8%)	12 324	1 337
Newspaper	31 (24.6%)	11 614	1 129
Oral	16 (12.3%)	26 875	1 017
Expository	10 (7.7%)	6 647	471
E-mail	16 (12.3%)	4 405	438
Fiction	8 (6.2%)	9 956	327
Political	3 (2.3%)	5 470	312
Technical	5 (3.8%)	2 517	101

Table 2. Language variety distribution

Origin	Text extracts	Words	NEs
Portugal	63	33 618	2 550
Brazil	60	42 073	2 274
Asia	3	2 864	233
Africa	3	1 253	75

(TEMPO), there were enough data to justify classification in the following categories: OBRA: titles (works of art, man made things), COISA: things (objects which have names), ACONTECIMENTO: events, and a further bag of others (VARIADO).

Figure 1 shows how the different kinds of NEs are distributed in the GC, while in Table 3 we present the quantitative distribution of the categories as far as size in words is concerned, using Linguateca's publically available tokenizer.

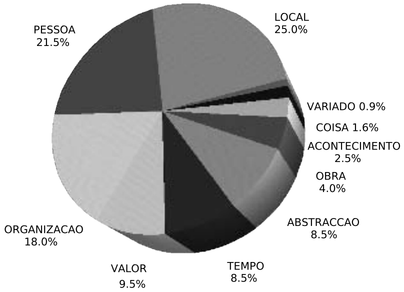


Fig. 1. Kinds of NEs in the GC

Table 3. Size in words of the NEs in HAREM's golden collection

Category	Average	Median	Standard Deviation
OBRA	3.34	2	2.94
ACONTECIMENTO	3.25	3	2.52
ORGANIZACAO	2.25	1	1.94
VARIADO	2.24	1	2.60
PESSOA	1.97	2	1.16
ABSTRACCAO	1.92	1	1.25
TEMPO	1.86	1	1.33
VALOR	1.77	2	0.96
LOCAL	1.67	1	1.50
COISA	1.24	1	0.59
TOTAL	**1.99**	**1**	**1.63**

Let us present some of the distinctions encoded in the types for each category (for further details, see the guidelines in Portuguese):

PESSOA. When NEs denote people, we make two kinds of orthogonal distinctions: One person vs. a group of people; and whether the person is described by its (own) name, by its job (title), or by its membership in a particular group (the metonymical use of companies or political movements). Observe that there are two kinds of groups: those who have a name as group, and those who are syntactically plural (pairs like *Romeu e Julieta* or plural CARGO (job/role) noun phrases, like *os Presidentes da Rússia e da China*, "Russia and China's presidents").

ABSTRACCAO. Abstract concepts turned out to constitute a sizeable proportion of capitalized entities in Portuguese, ranging from scientific areas to political programmes, to metonymical uses of an author name to refer to the collection of his works, etc. Also, the use of names as names (fairly common in all genres) was separately identified as a type of ABSTRACCAO called NOME.

ORGANIZACAO. Organizations were divided into companies, administration (government) and the rest (called institutions). Still, we found the need to independently classify sub-organizations, such as committees or departments, which were a kind of more flexible organization, but were neverthelesss felt to be different from groups of people without a goal as group.

ACONTECIMENTO. We distinguish three kinds of events: unique ones, which have a name identifying something that happened in a particular time, and other events which are often organized (and may be repeated) and encompass smaller (indivisible) events.

COISA. Objects can also be named, although, contrarily to people, there are more metonymical references than unique objects named. Although there are unique objects (type OBJECTO) that do have a name, like boats: *a fragata D. Luís*, most often – at least in technical domains – things are referred by the classes they belong to. An interesting distinction we catered for in HAREM is instance of a given kind (type CLASSE) and metonymical reference through the use of the same class name to denote an individual object (type MEMBROCLASSE).[9] Obviously, we only consider such cases as NEs when, because they are named after their discoverers, have to be spelled in capitals, like *aparelho de Golgi*, *constante de Planck*, *contador Geiger*. Avowedly an original decision, we expected systems to classify *de Golgi*, *de Planck* and *Geiger* alone (and not the full NPs) as COISAs of type CLASSE or MEMBROCLASSE, and not as PESSOA.

OBRA. Another category found to account for a lot of proper names in Portuguese text concerns works of art and other products of intellectual activity. An interesting difference emerged in the way unique works as opposed to reproducible are mentioned in Portuguese: *Guernica*, *Gioconda* or *a Torre Eiffel* do not take quotes, contrary to talk about books, musical pieces or movies.

LOCAL. Locations were divided into geographical landmarks and political or administrative locations.[10] Other kinds of locations identified and accomodated by the HAREM scheme are virtual locations (such as newspapers or the Internet), addresses (with a specific syntax of their own), and any named physical body which in a particular context is used for naming a place (like *Teatro São João*).

3.4 Vagueness and Alternative Delimitation

It is not unusual that classification of linguistic objects is not mutually exclusive: in fact, we believe that vagueness is a constitutive property of natural language [25] and that any NLP system should take it into consideration.

In addition, sometimes human annotators do not have enough information to decide, even consulting other knowledge sources, such as the Web: How many (Portuguese

[9] The difference between “I love Ferraris” or “my Ferrari is here”.

[10] Not to be confused with political actors represented by a place name, considered ORGANIZACAO in HAREM, as in *Os EUA não assinaram o tratado de Kyoto* (“USA did not sign the Kyoto treaty”). The distinction here is between human geography and nature, plausibly relevant for different applications.

speaking) readers know that *Miss Brown* is a Brazilian band or that *os pastéis de Belém* is often used as a place in Lisbon?

We therefore employed the disjunction operator (|) in both situations, abiding by two guiding principles:

1. We are dealing with real text and we are measuring systems' performances in real text. Therefore we should not exclude "too difficult" cases from the GR.
2. We do not want arbitrary decisions to be taken by annotators – if there is a ceiling, in the sense that humans cannot decide above it, it makes no sense to require that a machine should decide more fine-grainedly than humans [26].

The same principles are operational at the level of NE alternative segmentation as well, as in `<ALT> <ORGANIZACAO>Governo</ORGANIZACAO> de <PESSOA>Mário Soares</PESSOA> | <ORGANIZACAO>Governo de Mário Soares<ORGANIZACAO> </ALT>`.

Anyway, to avoid artificial proliferation of alternative NEs, we tried to separate the above cases where human readers had no clues to decide between different interpretations from those cases which could be easily formalized, and arbitrarily but consistently encoded. Examples of this latter are as following:

- encode only largest NE: if a title is followed by the name, mark only one person, as in *o Presidente do Parlamento Europeu Stefano Prodi*;
- do not practise recursive NE-labelling: place names in organization descriptions (as in *Câmara Municipal de Lisboa*), or organization names in title jobs (as in *Secretário-geral do PC do B*), are not tagged separately; [11]
- do not allow an arbitrary number of uncapitalized NE beginnings: we allow only titles and address forms (such as in *major Otelo* and *senhor Alves*) and disease hyperonyms (such as in *doença de Alzheimer*, and *sindroma de Down*);[12]

4 Discussion and First Results

No matter how simple the task to build a golden resource for NER may appear, many fundamental NLP issues had to be tackled. "Named entity" (or better, our Portuguese appropriation of it, *entidade mencionada*, literally "mentioned entity") did not have a consensual description, so we had to provide an operational definition for HAREM's purposes, which is obviously open to criticism.

One of these decisions, motivated by the need to compare fairly different systems was our definition of avowedly "non-standard" NEs like *de Planck*. However, this decision would not penalize systems that detected or classified whole terminological units, provided the systems included an "added-to-participate-in-HAREM"-rule, which would

[11] This is essentially linked to our wish to annotate in context, but one may later re-annotate the GC with some embedded NEs whose usefulness turns out to be consensual.

[12] This apparently unmotivated rule has a good explanation: if one was to mark **the entity** that included a named entity, too much dependence on systems' assumptions and complex parsing would occur. One might end up having to accept *a dona da barraquinha das bolas de Berlim* ("the owner of the candy shop selling Berlin balls", a kind of cake) or *o braço direito do rio Guadiana* ("the northern bank of the river Guadiana") as the right NEs.

propagate their original semantic classification to the new NE including only the capitalized sub-expression, but including preceding prepositions.

The most important decision taken, in any case, was our choice of disambiguation in context. There are two approaches to NE recognition, which are actually not specific of NER but of corpus annotation as a discipline, and have to do with how far one disambiguates or chooses a specific annotation value: Some researchers have opted to simplify NER by considering vague categories as final leaves in the classification, much in the same vein as early PoS ventures used so-called "portmanteau" categories, therefore failing to fully disambiguate the text [27]. We took the opposite path: We wanted to evaluate the task of selecting the (meaning of the) named entity in context. Accordingly, HAREM turned out to be more difficult than many of the tasks originally aimed by the participating systems.

Take the much discussed case of country names: to tag *Portugal* as a country requires a simple dictionary/gazetteer lookup, but to find places or political actions are two different goals, which we expect to be performed by different users with different user needs (and therefore possibly even implemented in different applications). That this is not an irrelevant theoretical concern, is evidenced by the real distributions for *Portugal*, *França* and *Brasil* in context in the GC (table 4):

Table 4. NE counts for single categories (normal) and vague categories (enclosed in parenthesis)

NE	LOCAL	ORGANIZACAO	PESSOA	ABSTRACCAO
	(administrative place)	(administration)	(group of people)	(ideal)
Portugal	32 (0)	16 (1)	0 (0)	0 (1)
Brasil	52 (1)	5 (2)	2 (0)	2 (1)
França	7 (0)	2 (0)	1 (0)	12 (0)

Table 5. Number of vague NEs, and number of alternative identifications, per category

Category	One category (<A>)	Vague class. (< $A_1 \| A_2$ >)	Vague ident. (<ALT>)
LOCAL	1227	44	10
PESSOA	972	59	30
ORGANIZACAO	895	64	13
VALOR	480	1	1
ABSTRACCAO	437	23	9
TEMPO	430	1	5
OBRA	194	22	4
ACONTECIMENTO	105	3	21
COISA	75	6	2
VARIADO	37	0	2

Incidentally, the NEs that had more different contextual interpretations in HAREM's GC were newspaper's names, employed in news text in many ways: as a person, representing the journalist that asks questions in an interview; as a place (of publication); as a group of reporters; as a company; or as a product (that for example sells well). Our GC – if representative enough – will also allow researchers to estimate relative probabilities of these different uses, in addition to provide them with contexts to discover discrimination patterns. In fact, one important advantage of evaluation contests, is the creation of consensual available resources, and we agree that "reusable test collections – which allow researchers to run their own experiments and receive rapid feedback as to the quality of alternative methods – are key to advancing the state-of-the-art"[28].

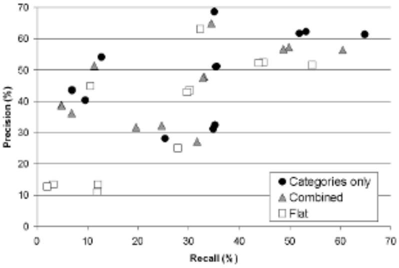


Fig. 2. Prec. vs Recall for Semantic Classification (Absolute scenario)

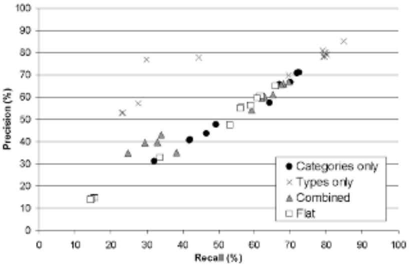


Fig. 3. Prec. vs Recall for Semantic Classification (Relative scenario)

Some idea of the difficulty of human annotation of the different categories in context can be obtained from Table 5.

We conclude this paper by providing an overview of HAREM results, using the present GC: Figures 2 and 3 depict systems' performance for semantic classification, according to two different scenarios: absolute (counting all NEs in the GC even if they had not been identified by the system), and relative (that is, relative only to correctly identified NEs). See [29, 30] for further details. We look forward to further studies in NER of Portuguese that employ this resource.

Acknowledgements

We are grateful to HAREM's organizing committee's other members and to all participants for feedback and enthusiasm, and especially to Cristina Mota for initiating the whole activity back in 2003. This work was supported by grant POSI/PLP/43931/2001 from Fundação para a Ciência e Tecnologia, co-financed by POSI.

References

1. Santos, D., Seco, N., Cardoso, N., Vilela, R.: HAREM: an Advanced NER Evaluation Contest for Portuguese. In: Proceedings of LREC'2006, Genoa, Italy (2006)
2. Hirschman, L.: The evolution of Evaluation: Lessons from the Message Understanding Conferences. Computer Speech and Language **12**(4) (1998) 281–305
3. Santos, D.: Avaliação conjunta. In Santos, D., ed.: Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa. (In press)
4. Santos, D., Barreiro, A.: On the problems of creating a consensual golden standard of inflected forms in Portuguese. In Lino et al, ed.: Proceedings of LREC'2004. (2004) 483–486

5. Santos, D., Costa, L., Rocha, P.: Cooperatively evaluating Portuguese morphology. In Nuno Mamede et al, ed.: Computational Processing of the Portuguese Language, 6th International Workshop, PROPOR, Springer Verlag (2003) 259–266
6. Grisham, R., Sundheim, B.: Message Understaning Conference - 6: A Brief History. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING-96), Copenhagen (1996) 466–471
7. Mota, C., Santos, D., Ranchhod, E.: Avaliação de reconhecimento de entidades mencionadas: princípio de AREM. In Santos, D., ed.: Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa. (In press)
8. Rocha, P., Santos, D.: CLEF: Abrindo a porta à participação internacional em avaliação de RI do português. In Santos, D., ed.: Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa. (In press)
9. Sang, E.F.T.K.: Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of CoNLL-2002, Taipei (2002) 155–158
10. Sang, E.F.T.K., Meulder, F.D.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proc. of CoNLL-2003, Edmonton (2003) 142–147
11. Lisa Ferro et al: TIDES 2003 Standard for the Annotation of Temporal Expressions. Technical report, MITRE (2004)
12. George Doddington et al: The Automatic Content Extraction (ACE) Program. Tasks, Data and Evaluation. In Lino et al, ed.: Proc. LREC 2004, Lisbon (2004) 837–840
13. Guthrie, L., Basili, R., Hajicova, E., Jelinek, F.: Beyond Entity Recognition - Semantic Labelling for NLP Tasks. In: Workshop proceedings, ELRA, Lisboa (2004)
14. Sekine, S., Sudo, K., Nobata, C.: Extended Named Entity Hierarchy. In González Rodríguez, M., Araujo, C.P.S., eds.: Proceedings LREC'2002, Las Palmas (2002) 1818–1824
15. Christian Bering et al: Corpora and evaluation tools for multilingual named entity grammar development. In Newman, S., Schirra, S.H., eds.: Proceedings of Multilingual Corpora Workshop at Corpus Linguistics 2003, Lancaster (2003) 43–52
16. Merchant, R., Okurowski, M.E., Chinchor, N.: The Multilingual Entity Task (met) overview. In: Proceedings of TIPSTER Text Program (Phase II), Tysons Corner, Virginia (1996)
17. Callmeier et al: COLLATE-Annotationsschema. Technical report, DFKI (2003) `http://www.coli.uni-sb.de/~erbach/pub/collate/AnnotationScheme.pdf`.
18. Arévalo, M., Carreras, X., Márquez, L., Martí, M.A., Padró, L., Simón, M.J.: A Proposal for Wide-Coverage Spanish Named Entity Recognition. Revista da SEPLN **1**(3) (2002) 1–15
19. Kokkinakis, D.: Reducing the effect of name explosion. In Guthrie, L., Roberto Basili, Eva Hajicova, Frederick Jelinek, eds.: Beyond Named Entity Recognition - Semantic Labelling for NLP Tasks. Pre-conference Workshop at LREC'2004, Lisboa, Portugal (2004) 1–6
20. Karlgren, J., Cutting, D.: Recognizing Text Genres with Simple Metrics Using Discriminant Analysis. In: Proceedings of COLING 94, Kyoto, Japan (1994) 1071–1075
21. Santos, D.: Towards language-specific applications. Machine Translation **14**(2) (1999) 83–112
22. Palmer, D.D., Day, D.S.: A Statistical Profile of the Named Entity Task. In: Proceedings of ANLP-97, Washington D.C. (1997) 190–193
23. Bick, E.: Multi-level NER for Portuguese in a CG framework. In Nuno Mamede et al., ed.: Computational Processing of the Portuguese Language, 6th International Workshop, PROPOR 2003, Springer Verlag (2003) 118–125
24. Mikheev, A., Moens, M., Grover, C.: Named Entity recognition without Gazetteers. In: Proceedings of EACL'99, Bergen (1999) 1–8
25. Santos, D.: The importance of vagueness in translation: Examples from English to Portuguese. Romansk Forum **5** (1997) 43–69

26. Calzolari, N., Corazzari, O.: Senseval/Romanseva: The Framework for Italian. Computers and the Humanities **34**(1-2) (2000) 61–78
27. Macklovitch, E.: Where the Tagger Falters. In: Proc. of the 4th International Coference on Theoretical amd Methodological Issues in Machine Translation, Montréal (1992) 113–126
28. Voorhees, E.M., Tice, D.M.: Building a Question Answering Test Collection. In Nicholas Belkin et al, ed.: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens (2000) 200–207
29. Cardoso, N.: Avaliação de Sistemas de Reconhecimento de Entidades Mencionadas. Master's thesis, FEUP, Porto, Portugal (2006) In preparation.
30. Seco, N., Santos, D., Cardoso, N., Vilela, R.: A complex evaluation architecture for HAREM. (This volume)

Functional Aspects in Portuguese NER

Eckhard Bick

Institute of Language and Communication,
University of Southern Denmark
eckhard.bick@mail.dk

Abstract. This paper describes and evaluates an improved and modified version of the PALAVRAS-NER parser, adapted for the HAREM joint evaluation task of Named Entity Recognition (NER) in February 2005. Apart from an extension to over 40 semantic categories, the system was changed from a lexeme-based to a token-based description, defining NER categories as functional and context-based rather than stable and lexematic. The Constraint Grammar rule body was changed accordingly, adding new rules and applying existing heuristic and disambiguation rules to contextual re-mapping of also lexically *known* material. In the joint evaluation, PALAVRAS-NER achieved the best overall results, achieving top ranks for both the identification, classification and morphology tasks.

1 Introduction and Previous Work

The PALAVRAS-NER parser is a progressive-level Constraint Grammar (CG) system, treating Named Entity Recognition (NER) as an integrated task of grammatical tagging. The original version, presented at the PROPOR 2003 (Bick 2003) and also used for Linguateca's avalia-SREC task 2003, implemented a basic tag set of 6 NER categories (person, organisation, place, event, semantic products and objects) with about 20 subcategories, following the guidelines of a joint Scandinavian NER project (*Nomen Nescio,* Johannessen et.al. 2005). Category tag candidates were added at three levels, and subsequently disambiguated by CG-rules:

(a) known lexical entries and gazeteer lists (about 17.000 entries)
(b) pattern-based name type prediction (morphological module)
(c) context-based name type inference for unknown words

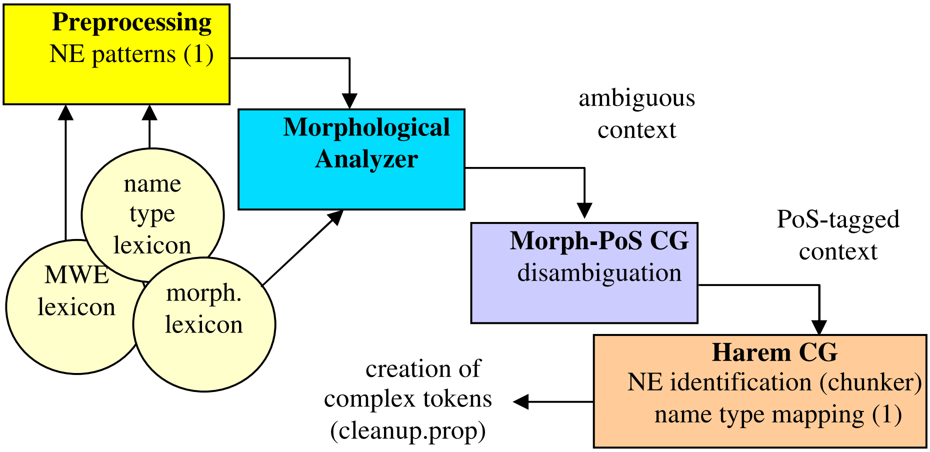
Since PALAVRAS originally was conceived primarily as a syntactic parser (Bick 2000), it fuses fixed expressions with non-compositional syntactic-semantic function into multi-word expressions (MWEs), creating complex tokens and in the process making life easier for the token-based syntactic CG-rules as well as avoiding arbitrary descriptive decisions as to the internal structure of such MWE[1]. Names, too, are treated as MWEs, and semantic NER-classes are assigned to the whole, not the parts.

[1] For corpus-users with a blank-space based token definition, MWEs can be unfolded and assigned an internal analysis by an add-on filter-program.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 80 – 89, 2006.

2 Recognizing MWE Name Chains

Illustration 1: Name chain identification modules



Identification of names, as a sequence of atomic tokens, was a separate task in the HAREM joined NER evaluation (www.linguateca.pt), and the PALAVRAS-system performed best, with an F-Score of 80.61%, in both the selective and total measures. Single-token names, with the exception of sentence-initial position, are clearly marked by upper case - therefore, since multi-token names can't be identified without chaining them into MWEs first, and since very few other (non-NE) cases involve productive MWE-chaining, the NE identification task is to a large degree identical to an MWE-recognition task[2]. The 2003 PALAVRAS-NER system (in this text, PAL-1), taking a more static approach, tried to fix MWE names *before* running the system's grammars - either by simple lexicon-lookup or by pattern-recognition in the preprocessor - and the only allowed post-grammar token alteration was fusion of adjacent name chains. This technique was replaced by a more dynamic, grammar based tokenisation approach in the new, 2005 system (henceforth, PAL-2), used for HAREM. Here, preprocessor-generated name candidate MWEs that cannot be verified in the lexicon as either known names or non-name polylexicals, are fed to the morphological analyser not as a whole, but in individual token parts, with < and > tags indicating start and stop of name MWE candidates. Thus, parts of unknown name candidates will be individually tagged for word class, inflexion and - not least - semantic prototype class. In addition, each part is tagged either @prop1 (leftmost part) or @prop2 (middle and rightmost parts). This technique has two obvious advantages over the old approach:

[2] Strictly speaking, the HAREM annotation and metrics did not employ MWEs per se, but rather XML-tags marking the start end end of name expressions. These XML tags were automatically added to PALAVRAS output before evaluation, at the same time turning semantic category tags into XML attributes.

1. It allows the morphological disambiguation grammar to establish the gender and number of names from their constituents, as well as internal morphological features, name-internal pp-constructions etc.
2. A specialized, newly-written name grammar can *change* the very composition of a name MWE, by removing, adding or replacing @prop1 start and @prop2 continuation tags.

For instance, the grammar can decide contextually whether sentence initial upper case is to be treated as a part of a name or not. Thus, certain word classes (prepositions, adverbs, conjunctions, finite verbs) can be recognized and tagged as no-name even with another upper case word to the right. Though a simple preprocessor might have sufficed to check for the closed classes, this is problematic due to ambiguity, and certainly not true of finite verbs, which are both open-class and often ambiguous with nouns, so the task has to be done after morphological analysis and disambiguation (illustration 1).

The name-chunker part of the Harem CG can progressively increase the length of a half-recognized chunk in a grammatically founded and context-sensitive way, for instance by adding conjuncts (e.g. the last two tokens in ... *Doenças Infecciosas e Parasitárias,* a1*)* or PPs (e.g. the last part of *a Câmara Municipal de Leiria,* a2). Since the parts of name chains at this stage are "perspicuous" as nouns or other word classes, valency potential may be exploited directly (a3). In the rules below, the MAP operator adds information (tags) to a TARGET for a given context (1 meaning "one word to the right", -2 "two words to the left etc.). BARRIER conditions can block a context if the barrier tag is found between the target and the context tag in question, while LINK conditions add secondary context conditions to an already instantiated context.

(a1)
MAP (@prop2) TARGET (KC) (-1 <prop2> LINK 0 ATTR) (1 <*> LINK 0 ATTR)
MAP (@prop2) TARGET <*> (0 ATTR) (-1 KC) (-2 <prop2> LINK 0 ATTR) ; where <*> = upper case, KC = coordinator, ATTR = attribute

(a2)
MAP (@x @prop2) TARGET PRP-DE (*-1 N-INST BARRIER NON-ATTR LINK 0 <prop1>) (1PROP LINK 0 <civ> OR <top>)
MAP (@x @prop2) TARGET PROP (0 <civ> OR <top>) (-1 PRP-DE) (*-2 N-INST BARRIER NON-ATTR LINK 0 <prop1>); where PROP = (atomic) proper noun, N-INST = nouns with a semantic-prototype tag of *institution,* <civ> = known *civitas* names, <top> = known place names, <prop1> = preprocessor-proposed start of name chunk.

(a3)
MAP (@prop1) TARGET <*> (0 <+a>) (1 PRP-A) (NOT -1 >>>) ; where <+a> = noun's or participle's binding potential for the preposition *a,* >>> = sentence start.

Not all name-part mapping rules are unambiguous - (a2), for instance, includes @x, meaning "wrongly assumed name part", along with @prop2, meaning "second part of name". Ultimately, a set of REMOVE and SELECT rules decides for each name part candidate if it is valid in context and if it is a first or later part of the chain. For instance, definite articles or the preposition "de" cannot be part of a name chain, if the token immediately to the right is not a second part candidate, or has been stripped of its name tag by another, earlier, rule:

```
REMOVE (@prop2) (0 <artd> OR PRP-DE LINK 0 @y) (NOT 1 @prop2)
```

The result, an unambiguous tag (@prop1=first part, @prop2=later part, @x=ex-name, @y=confirmed no-name) is implemented by a filter program, cleanup.prop, such that later programs and grammars will see only ready-made complex name tokens.

3 Semantic Typing of Name Tokens: Lexematic Versus Functional NE Categories

The next task, after identifying the name chain tokens, was to assign them a semantic category and subtype. The original PAL-1 did subdivide the 6 *Nomen Nescio* supercategories into subcategories, but recognized only about 17 partly experimental categories, while the new PAL-2 had to accommodate for HAREM's 9 categories and 41 subcategories. This meant more than doubling the category inventory, and category matching was in many cases complicated by the fact that matches were not one-to-many, but many-to-many. This difference was not, however, the most important one. Far more crucial, both linguistically (i.e. in terms of descriptive meaning) and application ally (i.e. in terms of parsing grammars), was the treatment of metonymy. For many name types, metonymy is a systematic, productive and frequent phenomenon - thus, author names may be used to represent their works, city names may denote soccer clubs and a country name may be substituted for its government. Here, PAL-1 subscribed to a lexeme based definition of name categories, while HAREM used a function-based category definition. In the former tradition, a given name would have one, unchanging lexematic category, while in the latter it would change category according to context. Thus, the name of a country would always be <civ> (civitas) in PAL-1, a hybrid category of place and organisation, allowing, for instance, both +HUM subject-hood, and BE-IN-LOC-adverbiality. According to the HAREM guidelines, however, hybrid categories were not allowed, and simply turning <civ> into <top> (place) would result in a considerable error rate in those cases, where the country-name *functions* as an organisation or a humanoid group, i.e. where it announces, suffers or goes to war. Likewise, institutions <inst> can be seen as both places and organisations, while the erstwhile <media> category implies a function-split between a newspaper being read (semantic product), burned (object) or sued in court (company). On the other hand, HAREM also introduced some distinctions that *were* lexematic rather than functional, for instance the split between the (money-making) *company* subtype and the non-profit *institution sub*type of the organisation category.

In order to handle the lexeme-function difference, PAL-2 had not only to increase its category inventory, but treat lexicon-, morphology- and pattern-derived categories as "potentialities" to a much higher degree than PAL-1 had done. 5 levels can be distinguished for such lexicon-dependence or -independence of name tagging:

1. lexicon-entered names that have a reasonably unambiguous name category (e.g. Christian names, to a lesser degree surnames, which can denote styles or an artist's collected work)
2. lexicon-entered names with semantically hybrid categories (<civ>, <media>, <inst>) or with systematic metaphoring (<brand> as <object>)
3. pattern/morphology-matched names of type (1)
4. pattern/morphology-matched names of type (2)
5. names recognized as such (upper case, name chaining), but without a lexicon entry or a category-specific pattern/morphology-match

Even in the PAL-1 evaluation (Bick 2003), where hybrid categories did not have to be resolved and where only few, strong rules were allowed to override lexicon- or gazeteer-supported name-readings (1. and 2.), this group had an error rate of 5%, indicating that for many names, ambiguity is not merely functional, but already hardwired in the lexicon (e.g. *Washington* as person or place name). In PAL-2, lexicon-derived categories were treated as contextual indications only, and the names carrying them were submitted to the same rule set as "unknown" names (3. - 5.), opening up for considerably more ambiguity and a correspondingly higher error risk.

Illustration 2: Name typing modules

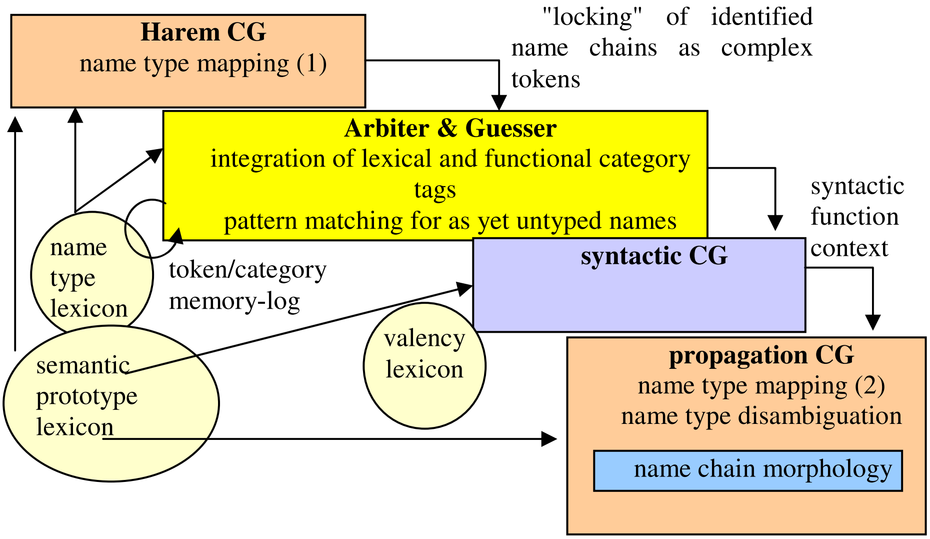

Harem CG
name type mapping (1)
"locking" of identified name chains as complex tokens
Arbiter & Guesser
integration of lexical and functional category tags
pattern matching for as yet untyped names
syntactic function context
name type lexicon
token/category memory-log
syntactic CG
valency lexicon
semantic prototype lexicon
propagation CG
name type mapping (2)
name type disambiguation
name chain morphology

Illustration 2 shows the distributed nature of PAL-2 and the interaction of its different name typing modules. An essential cut, the "locking" of identified name chains into complex tokens, is made between the (new) Harem CG on the one hand and the (modified) syntactic module and propagation CG on the other. While the former *(micromapping)* works on minimal tokens (name-part words) and can exploit their PoS, semantics and morphology, this is not any longer possible for the latter, which is geared for syntactic clarity and therefore works on whole name chunks, and uses syntactic function and structure to "propagate" information from the rest of the sentence onto nouns *(macromapping).*

3.1 Micromapping: Name Type Rules Based on Name Parts and Patterns

Many of the micromapper's rules map chunking information at the same time as classifier tags, like in the following rule which types known towns or countries (<civ>) or typical *noun parts* (N-CIVITAS) of unknown towns or countries as "administrative", if they fill the subject slot of a human-agent or experiencer verb (V-HUM).

MAP (@admin @prop1) TARGET <*> (0 <civ> OR N-CIVITAS) (*1 V-NONAD BARRIER CLB LINK 0 V-HUM) (NOT 0 <prop2>)

It is the first part of a complex name (@prop1) that will carry the classifier tag (@admin), and both tag types may be mapped ambiguously for later rule based disambiguation. Once output from the micromapper CG has been "frozen" into name chunks, the *Arbiter* module checks the result against lexical data and morphological patterns, adding pattern based classifier tags where no category has been mapped, or where tags are marked as unsafe (e.g. <hum?>) by the pre-CG inflexion and derivation analyzer. The Arbiter is the only part of the system that has a text-level memory - logging identified names and their types to resolve the classification of name abbreviations and the gender of person names. Thus, on a small scale, entity disambiguation is used for NE typing as suggested by Blume (2005).

The Pal-1 based morphological analyzer only treats numbers as NE material if they are part of a larger NE, e.g. time and place units, not when occurring as mere quantifiers, as in the HAREM categories of QUANTIDADE, CLASSIFICACAO and MOEDA. In Pal-2, it is the Arbiter's pattern-matching module, not the "character-blind" CG, who has to recognize such number expressions as names, as well as pre-classify them for later treatment in the CG macromapper.

3.2 Macromapping: Name Type Rules Based on Syntactic Propagation

Macromapping is an adapted PAL-1 module that adds name type tags to already-identified name chains by using a number of syntactic "propagation" techniques (described in Bick 2003), exploiting semantic information elsewhere in the sentence :

1. *Cross-nominal prototype transfer*: Postnominal or predicative names (NE @N<, PRP @N< + NE @P<, @SC, @OC) inherit the semantic type through of their noun-head
2. *Coordination based type inference:* Types are propagated between conjuncts, if one has been determined, the other(s) inherit the same type.
3. *Selection restrictions:* Types are selected according to semantic argument restrictions, i.e. +HUM for (name) subjects of speech- and cognitive verbs, +TIME is selected after temporal prepositions etc.

In Constraint Grammar terms, macromapping is as much a mapping technique as a disambiguation technique, as becomes particularly clear from method (3), where many rules discard whole sets of name type categories by targeting an atomic semantic feature (+HUM or +TIME) shared by the whole group.

4 Evaluation

Table 1. Global HAREM results for PALAVRAS-NER, semantic classification absolute/total (i.e. all NE, identified or not) combined metric for 9 categories and 41 subcategories (types)

PALAVRAS Subtype	**Category** (incidence)	**HAREM Subtype**	**F-Score** (precision - recall)		
			cat total	**cat/types total**	**identification**
hum	**hum** PESSOA 20.5 %	INDIVIDUAL	**67.4** 61.1-75.2 rank 1	**65.6** 59.3-73.4 rank 1	**65.0** 58.6-72.7 rank 1
official		CARGO			
member		MEMBRO			
groupind		GRUPOIND			
groupofficial		GRUPOCARGO			
grouporg		GRUPOMEMRO			
admin	**org** ORGANI-ZACAO 19.1 %	ADMINISTR.	**58.7** 53.3-65.4 rank 1	**50.0** 45.3-55.9 rank 1	**56.3** 51.0-62.7 rank 1
inst, party		INSTITUICAO			
org		EMPRESA			
suborg		SUB			
date	TEMPO 8.6 %	DATA	**75.5** 79.8-71.7 rank 1	**72.2** 76.1-68.7 rank 1	**73.5** 77.7-69.8 rank 1
hour		HORA			
period		PERIODO			
cyclic		CICLICO			
address	**top** LOCAL 24.8 %	CORREIO	**69.6** 75.1-64.8 rank 3	**64.3** 69.4-59.9 rank 4	**68.6** 74.1-63.9 rank 3
admin		ADMINISTR.			
top		GEOGRAFICO			
virtual		VIRTUAL			
site		ALARGADO			
product, V	**tit** OBRA 4.3 %	PRODUTO	**21.3** 22.3-20.4 rank 1	**16.5** 17.3-15,8 rank 2	**19.7** 20.6-18.9 rank 1
copy, tit		REPRODUZIDO			
artwork		ARTE			
pub		PUBLICACAO			
history	**event** ACONTE-CIMENTO 2.4 %	EFEMERIDE	**36.2** 28.9-48.6 rank 4	**30.8** 24.6-41.3 rank 4	**32.7** 26.0-43.8 rank 4
occ		ORGANIZADO			
event		EVENTO			
genre, brand, disease, idea, school, plan, author,abs-n.	**brand** AB-STRACCAO 9.2 %	DISCIPLINA, MARCA, ESTADO, IDEIA, ESCOLA, PLANO, OBRA, NOME	**43.1** 47.3-39.6 rank 1	**39.6** 43.3-36.4 rank 1	**41.4** 45.4-38.0 rank 1
object	**object** COISA 1.6 %	OBJECTO	**31.3** 25.4-40.7 rank 1	**31.2** 25.5-40.3 rank 1	**31.3** 25.4-40.7 rank 1
mat		SUBSTANCIA			
class, plant		CLASSE			
prednum	VALOR 9.5 %	CLASSIFICADO	**84.3** 87.0-81.7 rank 1	**82.5** 84.8-80.2 rank 1	**82.2** 84.8-79.7 rank 1
quantity		QUANTIDADE			
currency		MOEDA			

The complete HAREM evaluation computed a number of other metrics, such as text type dependent performance. PAL-2 came out on top for both European and Brazilian Portuguese, but in spite of its Brazilian-optimized lexicon and syntactic parser, it achieved a higher F-Score for the latter (60.3% vs. 54.7%), possibly reflecting socio-linguistic factors like the higher variation of person names in a traditional immigration country like Brazil, its Tupi-based place names etc. all of which hamper regular pattern/morphology-based name type recognition[3]. HAREM also had separate *selective* scores, where systems were allowed to compete only for certain categories and skip others. However, since PAL-2 competed globally in all areas, selective scores equaled total scores.

Another HAREM measure not presented in the overview table were relative performance, defined as category recognition measure separately for only those NEs that were correctly identified. Since this was not done by presenting systems with a ready-chunked ("gold-chunk-") corpus, but by measuring only against NEs correctly recognized by the system itself, PAL-2 had the relative disadvantage of being the best identifier and thus having to cope also with a larger proportion of difficult names than other systems, resulting in suboptimal rank performance.

Table 2. Relative HAREM performance of PAL-2

HAREM Category	***combined***		***per category***		***PAL-1 F-Score***
	Precision - recall	***F-Score (rank)***	***Precision-recall***	***F-score (rank)***	
PESSOA	90.1-91.9	91.0 (3)	92.7-94.0	93.4 (3)	92.5
ORGANIZACAO	77.0-79.0	78.0 (5)	91.1-92.4	91.8 (7)	94.3
LOCAL	87.7-89.3	88.5 (7)	96.1-95.5	95.8 (5)	95.1
OBRA (tit,brand,V)	58.5-59.5	59.0 (3)	75.3-76.6	76.0 (3)	ABSTRACT 84.3 (tit, genre,ling) OBJECT: 57.1 (brand,V,mat)
ABSTR. (genre,ling)	82.6-85.6	84.1 (1)	90.5-93.2	91.8 (1)	
COISA (brand,V,mat)	98.8-98.8	98.8 (1)	100-100	100 (1)	
ACONTECIMENTO	69.6-72.6	71.1 (5)	81.9-85.4	83.6 (5)	88.7
TEMPO	91.5-91.5	91.5 (4)	96.8-95.5	95.8 (5)	-
VALOR	94.2-95.8	95.0 (1)	96.6-97.6	97.1 (1)	-

For a direct performance comparison between PAL-1 and PAL-2, only the per-category scores are relevant, since even if subcategory scores had been available for PAL-1, score differences might simply reflect the difference in type set size. Even so, however, scores neither matched nor differed systematically. Of the major categories, *person* and *place* scored better in PAL-2/HAREM than what was published for the lexeme-based approach in PAL-1 (Bick 2003), while *organisation* and *event* had lower scores. Interestingly, the major categories (person, organisation, place) even *ranked* differently, with *person* higher (lowest in PAL-1) and *organisation* lowest (second in PAL-1). The reason for this may reside in the fact that the function of hu-

[3] Alas, since all HAREM participants but the winner were anonymous, and different code names were used for the Brazilian and Lusitan evaluation, this pattern could not at the time of writing be verified as either general or system-specific.

man names is much more likely to stick to its lexeme category, while organisations frequently *function* as either human agents or place names[4]. The abstract and object categories of PAL-1 were not directly comparable to the ABSTRACCAO and COISA categories of HAREM, since the latter also had OBRA, drawing (book etc.) titles from PAL-1's *abstract* category and brands (unless *functioning* as objects) from the *object* category, with a number of minor subcategories and function distinctions further complicating this 2-to-3 category match.

5 Conclusion: Comparison with Other Systems

Though state-of-the-art NER systems make often make use of lexical and grammatical information, as well as extra-textual gzaeteer knowledge, most do so in a framework of data-driven statistical learning, using techniques such as HMM, Maximum Entropy, Memory or Transformation-based Learning. The statistical learning approach has obvious advantages where language independence is desired, as in the CoNLL2002 and CoNLL2003 shared tasks (Tjong Kim Sang 2002 and 2003), but language-specific systems or subsystems may profit from explicit linguistic knowledge (hand-written rules or lexica), as e.g. in a number of Scandinavian NER systems (Bick 2004 and Johannessen et.al. 2005). Petasis (2004) describes a 4-language NERC system with hybrid methodology, where the French section relies on human modification of rules machine-learned from an human-annotated corpus. PALAVRAS-NER stands out by being entirely based on hand-written rules, both locally (morphological pattern recognition) and globally (sentence context) - not only in assigning the grammatical tags used as context by the NER-system, but also within the latter itself. However, though PAL-2's rule based method worked best in the Portuguese HAREM context, with overall F-Scores of 80.6 for identification and 63.0/68.3 for absolute/relative category classification, it is difficult to compare results to those achieved for other languages, due to differences in metrics and category set size. In the CoNLL shared tasks on newspaper-text, the best absolute F-scores were 88.8 (English), 81.4 (Spanish), 77.1 (Dutch) and 72.4 (German) for a 3-way category distinction: *person, organisation, place* (plus *miscellaneous*), and given PALAVRAS-NER's high *relative* scores for these categories (93.4, 91.8 and 95.8), its lower total scores may well be due to suboptimal identification, reflecting either shortcomings of the PAL-2 rule system in this respect or linguistic-descriptive differences between the gold-standard CD and PALAVRAS-NER[5]. However, it is not at all clear how the CoNLL systems would have performed on a large (41) subcategory set and HAREM style mixed-genre data[6]. On the other hand, HAREM's category-specific and relative rank scores clearly show that there is much room for improvement in Pal-2, especially for the place and event categories, where it didn't rank highest (table 1). Also, Pal-2 appears to be *relatively* better at name chunk identification than at classification, since

[4] The *commercial* vs. *administrative* distinction also increases PAL-2's error risk.

[5] Such differences are particularly relevant for a system built by hand, not from training data. Thus, PAL-1 made far fewer chunking errors when evaluated internally (Bick 2003).

[6] The MUC-7 MENE-system (Borthwick et.al. 1998), for instance, experienced an F-Score drop from 92.2. to 84.2 even within the same (newspaper) genre, when measured not on the training topic domain, but in a cross-topic test.

it ranked lower in the relative scores (on correct chunks only) than in the absolute scores (identification task included). However, improvements do not necessarily have to be Pal-2-internal: Given an integrated research environment and a modular perspective (for instance, a cgi-integrated web-interface), a joined Portuguese HAREM system could act on these findings by delegating the identification and classification tasks to different systems and by applying weighted votings to exploit the individual strengths of specific systems, thus seamlessly integrating rule based and statistical systems.

Acknowledgments

The authors would like to thank the Linguateca team for planning, preparing, organising and documenting HAREM, and for making available a multitude of evaluation metrics in a clear and accessible format.

References

1. Bick, Eckhard (2000). The Parsing System 'Palavras' - Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus University Press, Århus
2. Bick, Eckhard (2003). Multi-Level NER for Portuguese in a CG Framework. In: Nuno J. Mamede et.al. (eds.) Computational Processing of the Portuguese Language (Proceedings of PROPOR2003, Faro, June 26-27, 2003), pp.118-125. Springer
3. Bick, Eckhard (2004). A Named Entity Recognizer for Danish. In: Lino et al. (eds.), *Proc. 4th Int. Conf. on Language Resources and Evaluation, LREC2004 (Lisbon, 2004)*, pp. 305-308.
4. Blume, Matthias (2005). Automatic Entity Disambiguation: Benefits to NER, Relation Extraction, Link Analysis, and Inference. In: Proceedings of 2005 International Conference on Intelligence Analysis. https://analysis.mitre.org/proceedings
5. Borthwick, Andrew & Sterling, John & Agichtein, Eugene & Grishman, Ralph: NYU: Description of the MENE Named Entity System as Used in MUC-7. In: Proc. of the 7th Message Understanding Conf. (MUC7), April 29th - May 1st, Fairfax (1998)
6. Johannessen et.al. (2005). Named Entity Recognition for the Mainland Scandinavian Languages. In: *Literary and Linguistic Computing*, Vol. 20, No. 1, pp. 91-102. Oxford University Press
7. Petasis, P. et. al. (2004). "Adaptive, Multilingual Named Entity Recognition in Web Pages". In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), pp. 1073 – 1074, Valencia, Spain, August 22 – 27, 2004.
8. Tjong Kim Sang, Erik (2002). Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: *Proceedings of CoNLL-2002*, Taipei, Taiwan, 2002, pp. 155-158.
9. Tjong Kim Sang, Erik & De Meulder, Fien (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 142-147.

SIEMÊS - A Named-Entity Recognizer for Portuguese Relying on Similarity Rules

Luís Sarmento

Faculdade de Engenharia Universidade Porto (NIAD&R) & Linguateca (Porto Node)
las@fe.up.pt

Abstract. In this paper we describe SIEMÊS, a named-entity recognition system for Portuguese that relies on a set of similarity rules to base the classification procedure. These rules try to obtain soft matches between candidate entities found in text and instances contained in a wide-scope gazetteer, and avoid the need for coding large sets of rules by exploiting lexical similarities. Using this matching procedure, SIEMÊS generates a set of classification hypotheses based solely on internal evidence, which may be disambiguated in a later step by relatively simple rules based on contextual clues. We explain SIEMÊS architecture and its named-entity identification and classification procedure. We also briefly discuss the results of the participation of SIEMÊS in HAREM, the named-entity evaluation contest for Portuguese, and describe future work.

1 Introduction

The goal of Named Entity Recognition (NER) is to identify and classify entities that have a proper name, as for example people, places or organizations, and also important numeric or hybrid expressions, like references to time or values. Named Entity Recognition is important role in many NLP applications, such as for example Question Answering [1] or Machine Translation [2], because it provides a useful semantic enrichment of the text. Generally speaking, NER systems are usually composed by a set of analysis rules, and one or more gazetteers, containing names of people, organizations, place, etc. The rules coded in a NER system tend to exploit both internal clues (or surface clues) such as capitalization, or the structure of certain expressions, and external clues (or contextual clues) to perform the correct identification and classification of the candidates. Gazetteers, on the other hand, are used to lookup information that may help the identification and classification processes. To develop a complete NER system, both components (rules and gazetteers) are usually required. Developers may build their own gazetteers or rely on existing ones, which are sometimes developed by official institutions and made available to the general public. For developing the rule set, one can apply machine-learning algorithms to learn rules from a reference corpus [3] or, alternatively, develop and tune the rules manually. However, more often than not, the appropriate reference corpus is not available, so coding the rules manually becomes the only feasible option. In those cases, developers usually end up with a very large and complex set of rules that are difficult to maintain, debug and expand. In this paper we

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 90 – 99, 2006.

present SIEMÊS, a NER system for Portuguese that follows a strategy that greatly reduces the number of rules to be coded. SIEMÊS relies on a small set of high-level rules that are able to find similarities between the entities found in text and instances stored in a wide-scope gazetteer. By exploiting such similarities, we greatly reduced the need for explicitly coding large sets of specific rules. This strategy allowed us to develop from scratch in a few weeks one of the top scoring NER systems that participated in HAREM [4], the Portuguese NER Evaluation Contest held in 2005.

2 Addressing the Named-Entity Recognition Task

SIEMÊS was designed to identify and classify references to named-entities according to the particular sense in which they were mentioned. This is a much harder challenge than traditional NER task definition [5] because it involves determining the semantic role of the entity in that particular context. In such a classification scenario there are three distinct types of ambiguous cases that need to be addressed.

The first one occurs when the same name is (partially) taken by entities belonging to distinct classes. For example, it is common for people to have names of certain places (cities, regions, etc). This can be considered the classical case ambiguity in the task of NER, where different classes of entities share similar lexical representations. For example, "Lima" is both the name of a City and a frequent Portuguese / Spanish surname. The traditional NER task involves being able to decide if, in a given context, "Lima" refers to a place ("A conferência foi organizada em Lima" / "The conference was held in Lima") or to a Person ("A conferência foi organizada pelo Lima" / "The conference was organized by Lima"). We will refer to these cases as Type I Ambiguity, and it is closely related to homography.

The second type of ambiguity is subtler, and it requires determining how a given entity is being mentioned in context, i.e., which semantic role it is assuming. Although certain names can be assigned unambiguously to a certain precise class of entities (or even to a precise entity), the context in which the referenced entities are mentioned implies different semantic roles. Taking for example the case of "Museu de Serralves" / "Serralves Museum" in the two following situations: (i) "Encontrámo-nos no Museu de Serralves"/ "We met at Serralves Museum" and (ii) "Museu de Serralves adquire Picasso" / "Serralves Museum buys a Picasso". While in the first case "Museu de Serralves" is mentioned as a Place, in the second case it is mentioned as an Organization. We are obviously referring to the same entity but the semantic role it assumes is different in both cases. We will refer to these cases as Type II Ambiguity. In certain cases, the number of possible semantic roles that may be found in practice is very high. Type II Ambiguity has several practical implications when using NER systems during pre-processing stages of other applications. For instance, in question-answering applications recognizing this difference may be decisive in allowing the system to correctly find and answer to "where is?" (or "who is?") questions.

There is yet another case of ambiguity (Type III Ambiguity), which lies somewhere between the previous two. Due to strong relationships between certain pairs of entities it is very common to use the same name when referring to both. Disambiguation of which of the possible entities is being actually referred to depends almost entirely on the context. Using the last example, “Picasso” is not mentioned as a Person but as an Object (presumably a painting). This situation is often found associated to objects / consumer products and their creators / producers (e.g.: “Kispo”, “Coca-Cola”, a newspaper, etc...).

The diversity of situation makes the NER a very challenging task. In such a scenario virtually any NE found in text can be classified in several different ways. However, the working hypothesis we followed for developing SIEMÊS is that for many ambiguous cases, disambiguation usually involves deciding only between a few classification hypotheses, which can eventually be traced beforehand and treated separately by rules targeting those cases. We thus believe that classification in context (i.e. with full sense-disambiguation) becomes easier if we first generate a set of reasonable classification hypotheses disregarding the context, and then performing disambiguation over that smaller set of hypotheses. SIEMÊS was thus designed to perform the complete classification procedure in three consecutive stages:

1. identification of candidate boundaries: the goal at this stage is to identify token sequences that could be valid NE’s;
2. generation of classification hypotheses: in this stage SIEMÊS tries to obtain several (possibly ambiguous) classification hypotheses based solely on internal / surface evidence and on information stored in the gazetteer.
3. classification in context / disambiguation: in this step SIEMÊS tries to determine the classification in the context of the NE candidate based both on the classification hypotheses obtained in the previous step and on the analysis of the context surrounding the candidate.

We will explain each of these stages more thoroughly, but for example the sake of clarity let as assume that during Stage 2 (generation of classification hypotheses) SIEMÊS is able to discover by examining its gazetteer that a certain candidate could refer both to a place or to a person (Type I Ambiguity). This finding will only be considered a working hypothesis and will have to be disambiguated in Stage 3 by rules that specifically deal with possible Place-Person ambiguity. Or, going back to “Museu de Serralves” example (Type II Ambiguity), if SIEMÊS generates an hypothesis that a candidate could be classified as Location, such hypothesis will have to be confirmed (or changed) in Stage 3 by another set of rules that deal with this Location-Organization ambiguity. Interestingly, this is exactly the opposite strategy to that described in [5] where the NER system first analyzes contextual evidence, and only in a later stage it relies in internal evidence of candidates found.

3 Stage 1: Identification of Candidate Boundaries

Valid NE candidates are strings of tokens that have at least one numeric or uppercased token. In a first iteration, SIEMÊS starts by finding in text possible NE candidates, which are considered seeds. Numeric and alphabetic seeds are then treated separately. SIEMÊS starts by processing numeric seeds since they are not prone to severe ambiguity problems. A grammar of numeric expressions is then used to identify time expressions and also quantities such as money, speed, weight, etc. In the case of numeric expressions identification and classification are made in a single step, and the whole procedure is based on a grammar of numeric expression. A similar process is used for "hybrid" seeds that are also identified and classified in this first stage, namely those that are related to URL, email addresses and telephone numbers. The most important case is concerned with how SIEMÊS processes the alphabetic seeds. Alphabetic seeds are allowed to grow forward according to certain rules in an iterative process. In each iteration, identified seeds are allowed absorb the next words to the right if these words are either seeds or if they are considered valid connectors. A valid connector is simply a sequence of words (not usually capitalized) that may connect words inside a valid named entity. Valid connectors were compiled manually by simple observation of frequent cases. The following names exemplify some valid connectors: "José [da] Silva", and "Fundação [para a] Computação Científica Nacional". In certain contexts some of these connectors become invalid. Those cases are eventually identified and corrected in the next stages. The seed growing process stops whenever no more valid connectors are found and the next word is considered not to be a seed. Seed splitting is also possible in Stage 3 if SIEMÊS is not able to classify a given grown seed.

4 Stage 2: Generation of Classification Hypotheses

The goal of this stage is to formulate reasonable hypotheses for classifying the alphabetic candidates previously identified in Stage 1. Following the previous examples, during Stage 2 we aim to obtain information that "José da Silva" has a good chance of being a person and that "Fundação para a Computação Científica Nacional", has a great chance of being a reference to an organization. Some of this information may be directly found in gazetteers. However, one cannot expect to find information regarding every possible entity in gazetteers. One possibility for circumventing this limitation is to develop rules that analyze internal evidence of candidates and generate hypotheses accordingly. For example, candidates that start with "Fundação para" are most probably organizations. Similar rules may be coded for dealing with people's names, certain locations, etc. But in the long run this strategy becomes burdensome and difficult to sustain because it requires coding many alternative conditions. A more versatile approach can be followed by observing that in many cases one can draw conclusions about the possible class of a given unknown entity by identifying features that are similar to entities already known, even if we completely disregard the context. For example, when confronted with a string like "Satini GTI", most readers will probably admit that it looks like the name of a car because this name has similar features to other well known car names. This does not immediately imply that "Satini GTI" refers to a car, but without additional context it certainly becomes a good hypothesis. Therefore, instead of relying in a very comprehensive gazetteer or in a large set of explicitly coded rules, SIEMÊS adopts a higher-level strategy: it uses a small set of similarity rules and a wide-scope gazetteer to formulate judgments about the possible classes of a given name. The key point here is that SIEMÊS will use the examples stored in the gazetteers to obtain "educated guesses" about what type of entity a name can refer to, by comparing it to all the examples of names stored. This strategy moves the classification effort from hard-coded rules to similarity rules over the gazetteer. The gazetteer will then need to be representative (i.e. store many different examples) rather than comprehensive (store many examples) in order to allow covering many different possibilities.

4.1 REPENTINO – A Wide-Scope Gazetteer

For this purpose we used REPENTINO [7] a wide-scope gazetteer that was compiled mostly by extracting names from corpora and from content-specific web sites. Instances in REPENTINO (approx. 450k) are organized in very wide classification hierarchy that includes 11 top categories and 102 subcategories.

Table 1. The top of REPENTINO hierarchy

Top-Category	# Sub-Categories / #examples	Top-Category	# Sub-Categories / #examples
Abstractions	13 / 5,832	Paperworks	9 / 4,439
Art/Med/Com	9 / 15,358	Products/Brands	15 / 9,262
Events	8 / 25,424	Beings	6 / 287,707
Places	16 / 50,810	Substances	4 / 1,472
Nature	5 / 869	Others	6 / 1,809
Organizations	11 / 47,143	Total	102 / ~ 450k

We would like to point certain very unorthodox categories such as for example "Paperworks", which contains names, laws, decrees and of all sorts of bureaucracy that are frequent in journalistic text. These unorthodox categories, despite not being usually considered in the NER task, provide insight in identifying (or excluding) candidates in a specific NER scenario.

4.2 Similarity Rules

SIEMÊS tries to discover possible sub-categories for a given candidate using the information stored in REPENTINO. Interesting sub-categories are those for which REPENTINO stores examples that are similar to the candidate. In other words, if a candidate looks similar to many items stored in REPENTINO under a sub-category of "Events", SIEMÊS assumes that the candidate may belong to that same sub-category. Obviously, SIEMÊS may find more than one sub-category for given example, and that information is then taken into account during the disambiguation step in Stage 3. SIEMÊS uses 5 high-level similarity rules to compare a given candidate to the contents of REPENTINO and to obtain a list of possible sub-categories. Rules, from the most restrictive to the least, are:

1. Exact Match: a given candidate is exactly matched in REPENTINO. This is considered the highest level of similarity.
2. Same N words in the beginning: the candidate begins with similar words to examples stored in REPENTINO. SIEMÊS tries to obtain the longest possible match. Examples: "Universidade Federal do...", "Associação Regional de…".

3. Same N words in the ending: the candidate end with similar words to other examples contained in REPENTINO. Again, SIEMÊS tries to obtain the longest possible match. Examples: "… Ltd.", "... GTI", "... Corp."
4. Number of common N-Grams: if none of the previous conditions is met, SIEMÊS tries to match all possible intermediate n-grams that may be generated from the candidate (if a candidate is composed of more than 4 words). This rule generates several partial comparisons for each candidate.
5. Frequent word(s) in certain subclasses. This is the least restrictive rule: it tries to match the candidate with items in REPENTINO that share *any* word in common with it.

Rules 2 and 3 are intended to deal with highly regular cases that are very frequent in Portuguese. Rule 2 is especially suited to cover Organizations and Events with long names, while rule 3 aims at solving cases that have typified endings such as brands or company names. Rule 4 explores a different kind of regularity that is often found in titles (books, movies, computer games). We performed simple tests using this rule and it was rather surprising how it was able to deal rather well with some movie titles that *were not stored* in REPENTINO. Rule 5, tries do deal those cases that do not have enough regularity except for a word that is highly discriminative and which may occur anywhere in the candidate, such as for example "Intercooler", "Pentium".

Depending on which rule has been applied in matching similar items, each of the classification hypotheses found has a certain score given by a weighting function. These functions, which have been developed heuristically and tuned manually, consider several parameters such as: (i) the number of items found in REPENTINO that are similar to the candidate; and (ii) the number of different second-level categories in which similar items were found in REPENTINO.

For instance, Rule 5 starts by splitting the candidate NE into its single-word components and for each of those words, w_i, REPENTINO is queried to obtain information about the number of entities per subcategory that contain that w_i. We thus obtain a list of pairs (subcategory, # entities containing w_i). We will denote this value obtained from REPENTINO as REP(subcat, w_i) so that a possible result for w_i = "silva" would be:

- REP(Beings::Human , "silva") = 1031;
- REP(Organization::Company , "silva") = 96;
- REP(Place::Loose Address ,"silva") = 42;

The same process is repeated for all the words of named entity candidate (nec_x) about which we are trying to formulate a hypothesis. Then for each subcategory ($subcat_k$) in REPENTINO we calculate the following score to assess the "likeability" of the candidate nec_x being an instance of that subcategory:

$$S_{Rule5}(subcat_k, nec_x) = \frac{1}{length(nec_x)} \sum_{i=1}^{i=length(nec_x)} \frac{REP(subcat_k, w_i)}{REP(subcat_k, *)} \qquad (1)$$

We thus obtain a list of weighted classification hypotheses for nec_x, which are then passed to Stage 3 to be disambiguated according to contextual evidence.

5 Stage 3: Classification in Context / Disambiguation

During Stage 3, SIEMÊS tries disambiguate the classification hypotheses found in the previous stage. This is achieved mainly by using very simple rules that analyze the close vicinity of the candidate in order to choose among the possible hypotheses. There are two basic sets of rules:

1. Rules that try to disambiguate between two or more classification hypotheses obtained in Stage 2. Some of these rules also deal with cases that we know beforehand that may be ambiguous even if only one classification hypothesis has been found in Stage 2.
2. Rules that try to classify candidates for which no relevant classification evidence has been found in Stage 2 nor by the previous set of rules.

The two sets of rules usually consider only one or two words preceding the candidates. They test for the presence of certain prepositions or other function words in order to obtain information that helps disambiguation. This is certainly not a very sophisticated process and more complete rules are probably needed for dealing with more complex cases. However, one should take into account that some important information has already been collected during Stage 2, which ideally should be able to reduce the uncertainty related to this classification problem. The first set or rules is responsible for solving certain frequent ambiguous cases such for example as Place-Person ambiguity, for which the rule is simply:

– if the Top 2 classification hypotheses for a given candidate (found in Stage 2) are Place and Person, and in this particular context the candidate is preceded by “no”, “na” or “em”, then the candidate is tagged as Place, but if it is preceded by “o” or “a”, then tag it as Person. Otherwise, tag the candidate as the highest scoring classification hypothesis, as given by Stage 2

Equally simple rules exist for dealing with the Company-Brand/Product, and several Place vs. Other Categories cases. During the development of SIEMÊS we confirmed that there are innumerable cases of ambiguity related to the Place category. Obviously, for many of those cases we were not able to resolve ambiguity using these simple rules.

The second set of rules works more as a last attempt to classify candidates about which SIEMÊS was not able to find enough information in Stage 2. In such situations, SIEMÊS tries again to obtain some information from the close surroundings of the candidate as well has from some morphologic clues. In some later experiments we noticed that these rules were extremely noisy, especially when tagging texts other than “well-behaved” journalistic text.

6 Participation in HAREM

During HAREM we were able to submit two different runs (SIEMÊS_A and SIEMÊS_B) to be evaluated, with minor changes between them. The two runs obtained good overall relative results: SIEMÊS was ranked in the Top 3 systems, for both identification task (finding the correctly delimited candidates) and semantic classification tasks (assigning the correct semantic tag to candidates). The absolute results however were not very high for any of the participating systems, which reflects the difficulty of such a generalized NER task. Since the most significant part our work is related to the classification stage of SIEMES, we briefly comment on results obtained in the identification task and we will focus with more detail in the results regarding the semantic classification task.

6.1 Identification Task

In the identification task, both runs obtained Precision values of approximately 77% and Recall values around 84%, ranking 2 and 3 position. We have not yet made a thorough analysis of these results to check if difficult or ambiguous cases were correctly solved. From the detailed result report provided by the HAREM organization we were able to observe results are consistent along all categories, except for TIME and DATE, i.e. the numeric entities, for which the result are extremely poor. This clearly shows that the grammars we developed for this kind of expression are clearly not comprehensive enough. Since these entities are very common (nearly 18% of the total number of entities in the collection to be tagged), our overall result can be significantly improved by expanding the grammars for numeric expressions.

6.2 Semantic Classification Task

In the Semantic Classification task, SIEMÊS_A and SIEMÊS_B had again very close results ranking third and second. The values of Precision for SIAMES_A and SIEMÊS_B were 56.8% and 57.3% and Recall values were 48.7% and 49.6%, respectively. SIEMÊS_B (the second run) performed slightly better, which may be explained by the fact that we inserted a few hundred instances in REPENTINO after the first round that seem to have played an important role.

Table 2 shows the performance of SIEMÊS_B for each Category. For numeric categories, SIEMÊS_B performed very poorly. From a comparative point of view the best results were obtained for PLACES, WORKS and EVENTS. In the case of PLACES, this good result has clearly benefited from the large number of such items stored in REPENTINO. An important result from our point of view is the good performance of SIEMÊS in highly regular Categories such as Organization, Events and Abstractions. In these cases, SIEMÊS relied mostly on similarity rules and on large base of varied examples stored in REPENTINO. In the case of the category Person, results were not as good as we expected for several reasons. Despite the fact that REPENTINO stores many examples of names, SIEMÊS was not able to correctly detect or classify several references to Person because they were made using short-names and nicknames. Clearly, in those cases a deeper contextual analysis is need.

We still need to perform a thorough component analysis of the results to evaluate the exact contribution of the similarity rules in the overall result of SIEMÊS, especially in the harder cases. However, in our opinion, the results obtained in HAREM confirm that the approach we followed is not inferior to others and that a significant amount of work in coding rules can thus saved.

Table 2. Global performance of SIEMÊS_B for each semantic category in HAREM

Category	Rank	Precision (%)	Recall (%)	F-Measure
PERSON	4	65.29	52.20	0.5801
ORG	2	57.63	41.17	0.4803
TIME	4	55.81	61.35	0.5845
PLACE	1	64.09	6983	0.6683
WORKS	1	29.75	1196	0.1706
EVENT	1	47.26	43.05	0.4506
ABSTRACT	2	41.80	28.60	0.3396
THING	2	30.00	13.33	0.1846
VALUE	8	53.32	37.42	0.4398

7 Future Work

Current work is addressing the application of similarity rules to context analysis in Stage 3. As described before, disambiguation rules in Stage 3 were too simple to solve harder cases but developing a larger set of rules may become unsustainable in the long run, especially when considering such a generalized NER task. Going back to the “Lima” example (Section 2), which can also refer to a river in Portugal, our simple disambiguation rules would probably classify the occurrence in “A aldeia foi inundada pelo Lima” (“The village was flooded by Lima”) as Person, when it is obvious that the referenced entity is a river. Our preliminary experiments in exploiting the similarity among contexts [8], have showed us that such cases might potentially be solved by using a set of contextual similarity rules and a very large reference corpus (or document collection). Searching in a reference corpus for contexts similar to the one surrounding the occurrence of the ambiguous named-entity, may lead to valuable information for its disambiguation. A simple search on Google shows that most of the words that follow the context “foi inundada pelo” / “was flooded by “include the word “rio” / “river” and related words (“lago” / “lake”, “mar” / “sea”, etc…) as well as explicit references to other rivers (e.g.: “Rio Jaboatão”, “Rio Beberibe”, etc.). We strongly believe that it is possible to solve some of the harder ambiguous cases using information obtained by searching “similar” contexts in large-text bases and performing simple analogy reasoning.

8 Conclusions

By implementing 5 high-level similarity rules instead of larger set of lower-level rules, we were able to develop one of the top scoring NER submissions which participated in HAREM evaluation contest. Similarity rules exploit certain regularities that exist in names and allow SIEMÊS to generate a set of reasonable classification hypotheses. Such hypotheses are then disambiguated in later steps by very simple contextual rules that focus on frequent ambiguous cases. We believe that further improvement will be achieved by extending the usage of similarity rules to context analysis procedures.

Acknowledgements

This work was partially supported by grants POSI/PLP/43931/2001 and SFRH/BD/23590/2005 from Fundação para a Ciência e Tecnologia (Portugal), co-financed by POSI.

References

1. Mihalcea, R., Moldovan D.. Document Indexing Using Named Entities. In: Studies in Informatics and Control, vol.10, no.1, January 2001.
2. Babych, B., Hartley A.. Improving Machine Translation quality with automatic Named Entity recognition. In: EACL 2003, 10th Conference of the European Chapter. Proc. of the 7th Int. EAMT workshop on MT and other language technology tools. Budapest Hungary (2003) pp. 1-8
3. Erik F., Tjong K., De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proc. of CoNLL-2003; Edmonton, Canada (2003) pp. 142-147.
4. Santos D., Seco N., Cardoso N., Vilela R.: HAREM: An Advanced NER Evaluation Contest for Portuguese. In: Proc. 5th Int. Conf. on Language Resources and Evaluation, Genoa, Italy (2006).
5. Grishman, R., Sundheim, B. Message Understanding Conference - 6: A Brief History. In: Proc. Int. Conf. on Computational Linguistics, Copenhagen (1996) pp. 466-471.
6. Mikheev, A., Moens M., Grover C.. In: Named Entity Recognition without Gazetteers. Proc. of EACL'99, ACL, Bergen, 8-12 June, (1999) pp.1-8.
7. Sarmento, L, Pinto, A., Cabral, L.: REPENTINO – a Wide-Scope Gazetteer for Entity Recognition in Portuguese. In: Proc. PROPOR 2006 - Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada. Itatiaia, RJ Brasil (2006).
8. Sarmento, L. A expansão de conjuntos de co-hipónimos a partir de colecções de grandes dimensões de texto em Português. In: Actas da 1ª Conf. de Metodologias de Investigação Científica, Porto, Portugal (2006)

Tools for Nominalization: An Alternative for Lexical Normalization

Marco Antonio Insaurriaga Gonzalez[1,2], Vera Lúcia Strube de Lima[1],
and José Valdeni de Lima[2]

[1] PUCRS - Faculdade de Informática,
Av. Ipiranga, 6681 – Prédio 16 – PPGCC – 90619-900 Porto Alegre, Brazil
{gonzalez, vera} @inf.pucrs.br
[2] UFRGS – Instituto de Informática,
Av. Bento Gonçalves, 9500 – 91501-970 Porto Alegre, Brazil
valdeni@inf.ufrgs.br

Abstract. The recognition of morphological variation and conceptual proximity of the words is crucial for tasks where the lexical normalization is used, such as term generation and matching in an information retrieval environment. We present tools that automatically perform nominalization for lexical normalization in Portuguese. Comparing the effects of three alternative strategies (stemming, lemmatizing, and our proposal: nominalization), we demonstrate through an experimental evaluation that nominalization, as lexical normalization, contributes to the performance improvement in a probabilistic information retrieval approach for Portuguese.

1 Introduction

Morphological variation and conceptual proximity of the words can have a strong impact on the effectiveness of an information retrieval (IR) system [11]. Two main alternative processes for lexical normalization of index and query terms have been studied during the last decades: stemming and lemmatizing. They perform a many-to-one mapping where semantically related words (or variants of a word) are represented by the same string at both indexing and searching phases in an IR environment.

Lexical normalization depends on the recognition of the morphological variation and conceptual proximity of words. This recognition is especially crucial for languages with complex inflectional morphology [3, 9, 17, 19]. In this paper, we present tools for performing a new kind of lexical normalization in Portuguese – the nominalization – as an alternative for the usual strategies. By means of the nominalization process, nouns represent adjectives, adverbs, and verbs of the text. We have tested a probabilistic IR system with four alternatives for lexical normalization: (i) absence of this strategy, (ii) stemming, (iii) lemmatizing, and (iv) nominalization.

The rest of this paper is organized as follows: Section 2 presents works related to lexical normalization; Section 3 describes the nominalization process; Section 4 explains differences between nominalization and the usual lexical normalization; Section 5, the focus of this paper, presents our tools for nominalization; Section 6 describes an experimental evaluation of strategies for lexical normalization; and Section 7 presents concluding remarks.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 100 – 109, 2006.

2 Related Work

Lexical normalization may be achieved through conflation, which reduces different words (or variants of a word) to a single form inferring conceptual proximities from morphological similarities. Stemming [5, 14] is a process of lexical normalization which conflates words with morphological similarities into a common representation: the stem, i.e., the common part of those words. This is the most usual IR term normalization strategy. However, while simple stemmers are adequate for languages such as English, more sophisticated strategies are demanded for languages with complex inflectional morphology like French, Finnish, German, Spanish, Turkish, and Portuguese. Therefore, stemmers for such languages present higher computational cost [19].

According to Krovets [11], stemmers do not take into account the differences caused by a word's meaning, and usually operate in the absence of any lexicon at all. However, according to Sever and Bitirim [17], in Turkish the selection of the root from a lexicon, as the stem of a word, increases the number of errors and, in some cases, it is not possible to choose the good stem without examining the context in which the word appears. However, Braschler and Ripplinger [3] demonstrate that stemming is useful for IR in German. They find that, compared to others where stemming is absent, such methods increase precision and recall.

Another usual term normalization process is lemmatizing [1, 10]. This process reduces the variant forms of a word to their canonical form (lemma), i.e., it reduces verbs to infinitive, and other words to their singular and (if it exists) masculine form. Korenius et al. [10] conclude that lemmatization is a better process than stemming for lexical normalization for Finnish, when documents are clustered for IR. Mayfield and McNamee [13] demonstrate that selection of a single n-gram, as a pseudo-stem, could be an effective and efficient language-neutral approach for lexical normalization in IR. However, according to Kettunen et al. [9], it is not obvious to decide which process to adopt for highly inflectional languages.

On the other hand, confirming the importance of nouns as concept descriptors, Lapata [12] treats the ambiguity of a particular class of compounds in English, where the head noun is derived from a verb. The correct interpretation of such compounds is important for IR because it helps ranking relevant documents. For instance, in the query *tratamento de cancer* (cancer treatment), when recognizing the derivation *tratar* (to treat) → *tratamento* (treatment) it is possible to retrieve documents in which *cancer* appears as the object of the verb *tratar*.

3 Nominalization as a Strategy for Lexical Normalization

Nominalization is an alternative strategy for lexical normalization based on the fact that nouns are usually the most representative words describing the document content [21]. Queries are usually formulated through noun phrases, as well. A noun phrase is a word group with syntactical behavior of subject, direct object, and (if preceded of a preposition) noun adjunct or indirect object [15].

In a broader context, nominalization is a word formation process in which a noun is derived from an existent word in the lexicon, which is mainly a verb or an adjective. In our work, nominalization is understood as the transformation of a word (adjective, verb, or adverb) found in the text, into a conceptually close noun existent in the lexicon. So, the derivation *amigavelmente* (friendly) → *amigo* (friend) is a valid form of nominalization, although in this case the adverb is already a derived word.

In our proposal for lexical normalization, nominalization operations obtain abstract and concrete nouns. Abstract nouns refer to events, qualities, states or other abstract entities, which can be obtained from adjectives, adverbs, or verbs, e.g.: *bom* (good) → *bondade* (goodness), *livremente* (freely) → *liberdade* (freedom), and *encontrar* (to meet) → *encontro* (meeting). Concrete nouns, on the other hand, refer to agents and are mostly obtained from verbs, or refer to something involved or associated with an entity, mainly obtained from adjectives, e.g.: *construir* (to build) → *construtor* (builder), and *numérico* (numerical) → *número* (number).

Nominalizations of verbs usually produce both abstract and concrete nouns, e.g.: *saltar* (to jump) → *salto* (jump), *saltador* (jumper). Nominalizations of adjectives and adverbs usually produce only abstract nouns, e.g.: *jovem* (young) → *juventude* (youth), and *facilmente* (easily) → *facilidade* (easiness). However, some adjectives and adverbs may have concrete nominalization only, e.g.: *fluvial* (fluvial) → *rio* (river). Finally, some words produce no new related nouns, such as *formidável* (outstanding). In this case, the lemmas of those words are considered the result of the lexical normalization.

4 Morphological Variation and Conceptual Proximity

Lexical normalization depends on the recognition of two linguistic phenomena: morphological variation and conceptual proximity of words. The morphological variations may be inflectional or derivational. The former do not affect word type, such as *entrar* (to enter) and *entrou* (entered). Derivational variation may affect word type, such as *limpar* (to clean) and *limpeza* (cleaning). Morphological similarity may cause conceptual proximity however there are words which are conceptually close and are not morphologically similar, such as *cair* (to fall) and *queda* (fall).

The recognition of these phenomena is crucial for term selection and matching in IR. Nominalization considers morphological variations but also takes into account conceptual proximities. So, for example, *biblioteca* is considered a concept different from *bibliotecário*. A lemmatizer would also obtain two terms from these nouns, but a stemmer would produce the same stem *bibliotec* here.

A stemmer may obtain the same stem from variants of a word or from words of different types. For example, *comércios* (commerces), *comercializou* (commercialized), and *comercial* (commercial) have the same stem *commerc*. On the other hand, lemmatizing does not alter word types. The words in the example have the lemmas *comércio* (commerce), *comercializar* (to commercialize), and *comercial* (commercial), respectively. From those referred words, just like stemming, nominalization also produces a unique term. Here, both stemming and nominalization treat the morphological variation, respectively through the stem *commerc* or through the noun *comércio* (commerce). These lexical normalization processes, unlike lemmatizing, recognize the conceptual proximity of all those words.

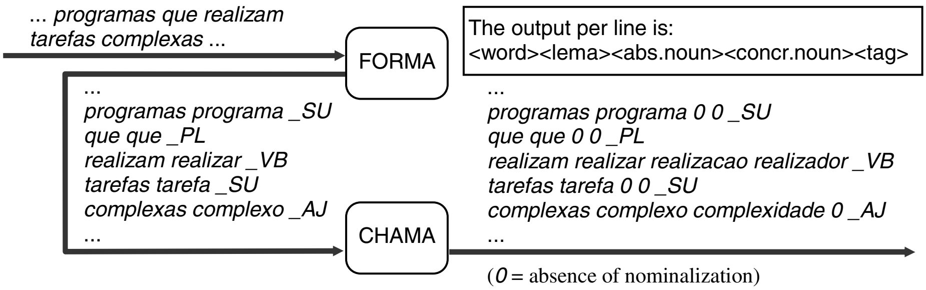
The nominalization of a verb usually obtains abstract and concrete nouns. For example, the verb *duvidar* (to doubt) has two relative nouns: *dúvida* (doubt) and *cético* (skeptic). This derivation is useful for term dependence models where events relative to verbs and their agents have important roles in text representation [6, 7].

Concerning the conceptual proximity, nominalization performs a kind of synonymy process in some cases. For example, the nominalization of *lindo* (beautiful), *bonito* (beautiful), and *belo* (beautiful) produces the same noun: *beleza* (beauty). However, there are three lemmas and three stems for those words. Stemming and lemmatizing infer conceptual proximities from morphological similarities, but this strategy does not cover all the possible situations. Nominalization also uses morphological similarities but recognizes many exceptions, such as *equino* (equine) → *cavalo* (horse).

These nominalization features enrich the lexical normalization in an IR system.

5 Tools for Nominalization

We developed a tool for nominalization (CHAMA), which reads as an input a lemmatized and tagged text and a complementary tool (FORMA) for lemmatizing and tagging. Figure 1 presents this strategy applied to an input text "… programs that perform complex tasks …".



... programas que realizam
tarefas complexas ...
FORMA
The output per line is:
<word><lema><abs.noun><concr.noun><tag>
...
programas programa _SU
que que _PL
realizam realizar _VB
tarefas tarefa _SU
complexas complexo _AJ
...
CHAMA
...
programas programa 0 0 _SU
que que 0 0 _PL
realizam realizar realizacao realizador _VB
tarefas tarefa 0 0 _SU
complexas complexo complexidade 0 _AJ
...
(0 = absence of nominalization)

Fig. 1. Strategy for nominalization: an input-output example

FORMA and CHAMA tools use finite automata that look for the largest character substring of the input string. We classify these automata in two types: left-right and right-left. A left-right automaton reads the input string starting from the leftmost character. A right-left automaton is based on suffix and starts from the rightmost character.

In general, these automata work as follows. Given an automaton *A* and an input string *t*, the reading of *t* by *A* ends (i) if *t* has been entirely consumed or (ii) if there is no valid transition for the next character of *t*. Then, the substring *st* read of *t* is accepted if the current state of *A* is a final state, otherwise *t* is rejected. An accepted substring *st* may be the root (in a left-right automaton) or the suffix (in a right-left automaton) of *t*, but other parts may be present in *st*. The objective of these automata is to read the largest substring that identifies *t*.

Each final state of the automata used here specifies derivation operations of the input string in order to obtain the necessary output (lemmas or nouns).

5.1 FORMA

FORMA performs lemmatizing and morphological tagging for words or punctuation marks. The tag set adopted is: *_AD* and *_AI* (definite and indefinite articles), *_AJ* (adjective), *_AP* (participle), *_AV* (adverb), *_CC* and *_CS* (coordinate and subordinate conjunctions), *_IN* (interjection), *_NC* and *_NO* (cardinal and ordinal numbers), *_PS*, *_PD*, *_PI*, *_PL*, and *_PP* (possessive, demonstrative, indefinite, relative, and personal pronouns), *_PN* (punctuation), *_PR* (preposition), *_SU* (noun), *_VA* (auxiliary verb), *_VB* (verb), and *_VG* (comma, parentheses, dash).

FORMA is a probabilistic tool which uses auxiliary data sets that were created from a lemmatized-tagged training corpus (the full text of the reference collection Folha94 described in Section 6.1). The auxiliary data sets constitute three automata for compounds, accents, and suffixes, and two probabilistic matrices *BP* and *AP*.

The automaton for compounds (with 577 compounds) is a left-right automaton that detects multi-word units, i.e., compound prepositions and compound adverbs. The automaton for accents (with 302 accentuated words) is a left-right automaton that detects words where the accent is a distinctive feature (which is important when dealing with Portuguese). The automaton for suffixes (with 43,476 entries) is a right-left automaton which analyses word suffixes for determining lemma and tag probabilities.

While the three automata obtain lemmas and tag probabilities, the matrices *BP* and *AP* estimate the probability of occurrence of a tag concerning the text. The matrix *BP* = $\{bp_{\text{mj}}\}$ is a 21x21 matrix where each element is $bp_{\text{mj}} = \Pr(j \mid m)$, i.e., the probability of occurrence of the tag *j* given a prior occurrence of the tag *m* in the corpus. On the other hand, $AP = \{ap_{\text{nj}}\}$ is a 21x21 matrix where each element is $ap_{\text{nj}} = \Pr(j \mid n)$, i.e., the probability of occurrence of the tag *j* given a posterior occurrence of the tag *n* in the corpus.

Figure 2 shows the strategy of FORMA tool, which reads the text (…programs that perform complex tasks…) and puts one token per line with lemmas (program, that, to perform, task, and complex) and morphological tags.

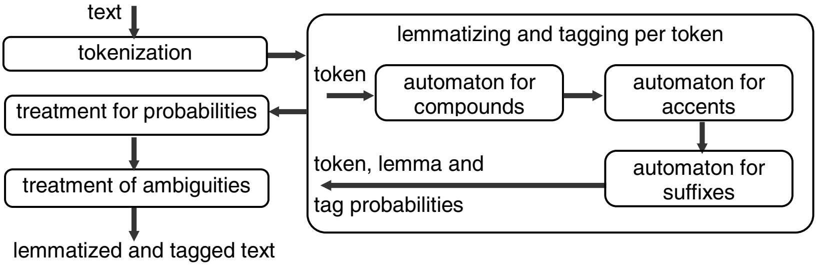

text
tokenization
treatment for probabilities
treatment of ambiguities
lemmatized and tagged text
lemmatizing and tagging per token
token
automaton for compounds
automaton for accents
automaton for suffixes
token, lemma and tag probabilities

Fig. 2. Strategy of FORMA tool

FORMA performs the following tasks (see Figure 2):

- Tokenization. Words and punctuation marks are detected. Also, from word contractions and combinations each component is derived as an individual token.
- Lemmatizing and tagging per token. This procedure for a token t_i may produce two types of output. The output type 1 is

  $$t_i \; L_i \; e_i$$

  where L_i is the final lemma for t_i (considered a tagged token at this moment), and e_i is the final tag (with 100% of probability) for t_i. The output type 2 is:

  $$t_i \; L_{0i} \; pe_{0i} \; L_{1i} \; pe_{1i} \; L_{2i} \; pe_{2i} \; ... \; L_{19i} \; pe_{19i} \; L_{20i} \; pe_{20i}.$$

  where pe_{ji} is the probability of tag e_j for t_i, L_{ji} is the lemma related to tag e_j for t_i (if $pe_{ji} = 0$, L_{ji} is an empty string), and t_i is considered an untagged token.

  The automaton for compounds reads sequences of 4 tokens at a time. An entry is accepted when the largest compound is derived as a new token and the output type 1 is produced. The automaton for accents reads only untagged tokens and, for each accepted entry (an accentuated word), the output type 1 is produced. Finally, the automaton for suffixes reads only untagged tokens as well, and for each accepted entry the output type 2 is produced.
- Treatment for probabilities. Given a sequence of 3 tokens $t_{i-1}t_it_{i+1}$ that includes the untagged token t_i, the new probability pe_{ji} for the tag e_j of t_i is:

  $$pe_{ji} = (\, b + 3pe_{ji} + a \,) \, / \, 5$$

  where $b = bp_{mj} \in BP$ if m is the final tag of t_{i-1}, otherwise $b = 0$; and $a = ap_{nj} \in AP$ if n is the final tag of t_{i+1}, otherwise $a = 0$.

  At the end, the tag with highest probability and the relative lemma are applied to t_i.
- Treatment of ambiguities. Ambiguous words, such as the verb form *foi* (went / was), are correctly tagged through the verification of the close tokens.

5.2 CHAMA

CHAMA is a tool that automatically obtains nominalized terms from the words in a Portuguese text. Figure 3 presents the nominalization strategy adopted for reading a lemmatized and tagged adjective, adverb, or verb, and obtaining the associated abstract and concrete nouns.

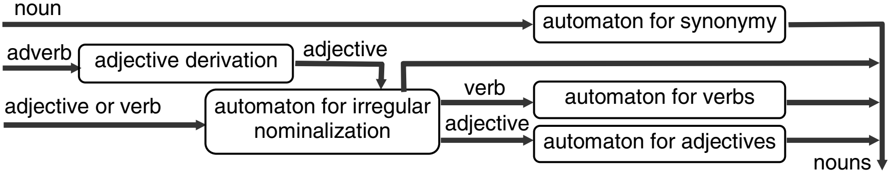


Fig. 3. Strategy of CHAMA tool for each word

Four finite automata perform the nominalization process: the synonymy automaton (SyA) with 327 entries, the exception automaton (ExA) with 4,223 exceptions, the adjective pattern automaton (AjA) with 663 patterns, and the verb pattern automaton

(VbA) with 351 patterns. SyA and ExA are left-right automata, while AjA and VbA are right-left ones. The rules for the derivations used by these automata were manually constructed following the descriptions of the Aurélio Portuguese Dictionary [4].

Each input word is processed according to its type:

- Preferred forms (for the dictionary [4]) are obtained from some nouns through SyA, such as *ruptura (rupture / break) → rompimento (rupture / breaking).*
- Adjectives are obtained from adverbs, such as *habilmente (skilfully) → habil (skilful).*
- For each adjective or verb, ExA is activated. If ExA accepts the input word, the nominalization occurs, such as *oriundo (proceeding) → origem (origin).*
- If the ExA rejects the input word, a pattern automaton (AjA or VbA) is activated. Then, the suffix of the input word determines the pattern that must be applied, such as **eletrico → *eletricidade,* for *elétrico* (electric), *fotoelétrico* (photoelectric), etc.

Each final state of the automata specifies nominalization operations in order to obtain abstract and concrete nouns.

6 Experimental Evaluation

6.1 Data and Methodology

We have used the document collection called Folha94 constituted by articles extracted from 229 editions of Folha de São Paulo newspaper of the year 1994. The 4,156 documents of Folha94 are originated from a subset of the corpus Mac-Morpho [8]. This corpus was lemmatized and tagged through the parser PALAVRAS [2].

In this evaluation, we followed the method in use by the Text Retrieval Conferences [20]. We took the title of 50 topics for generating test queries. Description and narrative of these topics, as well as the pooling method were adopted to obtain the relevance judgments with the top 100 documents for each evaluated strategy.

All the examined strategies use the conventional Okapi BM25 model [18] for probabilistic IR approach. We have comparatively evaluated four strategies: (i) Baseline VR strategy, which does not use lexical normalization (the terms are variant forms, i.e., the original words from the texts); (ii) ST strategy, which uses stemming through an adapted version of Orengo and Huyck stemmer [14]; (iii) LM strategy, which uses the original lemmas of Folha94; (iv) NM strategy, which uses nominalization.

NM uses the original lemmas and tags of Folha94. At the searching phase NM uses FORMA for lemmatizing and tagging. At both indexing and searching phases, CHAMA is used for nominalization. LM uses FORMA at searching phase. Concerning the query terms, lemmas and tags present in Folha94 are the same of those obtained by FORMA.

6.2 Results for Lexical Normalization

We have evaluated the lexical normalization and tagging processes used in this experiment. Table 1 shows the results of this evaluation. We examined samples extracted from the Folha94 texts for tagging (in corpus), stemming, lemmatizing (in corpus), and nominalization. For lemmatizing and tagging by FORMA, the samples were the full text of dissertation abstracts of computer science area because FORMA used Folha94 as trainning corpus.

Table 1. Lexical normalization and tagging evaluation

	sampling			
	sample size (# words)	error at 95% confidence level	precision	precision in the query terms
tagging (in corpus)	1,927	-0.006 to +0.004	0.989	–
tagging (by FORMA)	2,247	-0.011 to +0.009	0.949	0,980
stemming	1,003	-0.020 to +0.017	0.911	0,941
lemmatizing (in corpus)	1,319	-0.005 to +0.002	0.997	–
lemmatizing (by FORMA)	2,247	-0.008 to +0.006	0.974	0,990
nominalization	1,175	-0.008 to +0.004	0.991	0,980

Table 2 shows the size of the inverted files, the number of terms, and the time (in a 866 MHz Pentium III machine) spent, in average, at indexing and searching phases by each strategy. The indexing time considers the indexing of a document with 1,000 tokens, and the searching time takes into account the processing of a query with 2 terms.

Table 2. Memory space, terms, and processing time

strategies	inverted files (Kb)	# of terms	indexing time (s)	searching time (s)
VR	3,673	57,366	0.053	0.140
ST	2,873	24,013	0.062	0.137
LM	3,222	37,267	0.072	0.125
NM	3,713	36,479	0.105	0.130

The ST strategy presents the largest economy in memory space and in number of terms. LM and NM are similar in quantity of terms, although sometimes the nominalization process may index two terms (abstract and concrete nouns) obtained from a unique word. These duplications are, in general, compensated by reductions, when a single nominalized term is obtained from different words, such as *preferência* (preference) obtained from the adjectives *preferencial* (preferential) and *preferível* (preferable). In this case the lemmatizing process obtains two terms.

Concerning the searching time, NM is faster than ST. The main reason for this is that NM is more selective and retrieves less documents. The classification time for the retrieved documents of ST is longer than the classification time spent by NM.

6.3 IR Results

Figure 4 presents the recall-precision curves for the four strategies examined here. The precision values of LM and NM are similar with recall of 0.0 through 0.3. However, NM has the higher precision values with recall of 0.3 through 1.0. With recall of 0.4 through 0.7, the precision values of ST and LM are similar. Before recall 0.4, LM has precision values higher than ST. On the contrary, after recall 0.7, the inverse situation occurs.

Table 3 presents some values and differences of precision (P), recall (R), and mean uninterpolated average precision (MAP) measures for each strategy. The Wilcoxon signed rank test was employed with significance threshold 0.05. In Table 3 the statistically significant performance improvements and degradations are boldfaced.

Table 3 shows that NM has statistically significant differences relative to baseline VR strategy more than ST and LM. NM presents statistically significant improvement concerning one measure (precision values at 10 documents) relative to ST, and two measures (recall at 100 documents and MAP) relative to LM. Taking into account all measures of Table 3, LM and ST do not present statistically significant differences.

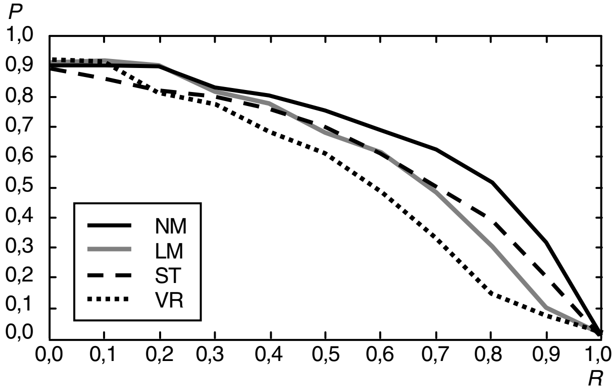


Fig. 4. Recall-precision curves

Table 3. Some values and differences of precision (P), recall (R), and MAP measures

	values				differences					
	VR	ST	LM	NM	ST–VR	LM–VR	LM–ST	NM–VR	NM–ST	NM–LM
P at 1 doc	0.920	0.900	0.920	0.900	-0,020	0,000	0,020	-0,020	0,000	-0,020
P at 10 docs	0.600	0.626	0.640	0.692	0,026	0,040	0,014	**0,092**	**0,066**	0,052
R at 100 docs	0.808	0.934	0,881	0.953	**0,126**	**0,073**	-0,053	**0,145**	0,019	**0,072**
MAP	0.631	0.741	0.732	0.789	**0,110**	**0,101**	-0,009	**0,158**	0,048	**0,057**

7 Conclusion

In this article we present two tools for implementing an automated process for nominalization: FORMA, which is an auxiliary tool for lemmatizing and tagging; and CHAMA, which is the nominalization tool in fact. Nominalization is presented as an alternative for the lexical normalization in Portuguese. This process recognizes both morphological variation and conceptual proximity of the words. This recognition, besides the reduction of the term number and the index file size, allows the better term matching at searching phase.

We have comparatively evaluated three strategies for lexical normalization in a probabilistic IR environment: stemming, lemmatizing, and nominalization. Our experiments demonstrate that nominalization may contribute to the improvement of the retrieval performance, as an interesting alternative for stemming and lemmatizing.

Nominalization may have an important role also in other applications that include natural language processing, like text categorization. The text representation through nouns may be a valid strategy because, considering adjectives, adverbs, and verbs, nouns carry more meaning and may well represent those words.

References

1. Arampatzis, A. T.; Weide, T. P.; Koster, C. H. A.; Bommel, P. Linguistically-motivated Information Retrieval. Encyclopedia of Library and Inf. Science, (2000) 69:201-222
2. Bick, E. The Parsing System Palavras, Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. A. University Press (2000)
3. Braschler, M.; Ripplinger, B. How Effective is Stemming and Decompounding for German Text Retrieval? Information Retrieval Journal, (2004) 7:291-316
4. Ferreira, A. B. H. Dicionário Aurélio Eletrônico – Século XXI. Nova Fronteira S.A., Rio de Janeiro (1999)
5. Frakes, W. B., Baeza-Yates, R. Information Retrieval: Data Structures and Algorithms. Prentice-Hall, New York (1992)
6. Gonzalez, M.; Lima, V. L. S. de; Lima, J. V. de. Binary Lexical Relations for Text Representation in Information Retrieval. 10th Int. Conf on Applications of NL to Inf. Systems, NLDB. LNCS, 3513. Springer, (2005) 21-31
7. Gonzalez, M. Termos e Relacionamentos em Evidência na Recuperação de Informação. PhD thesis, Instituto de Informática, UFRGS (2005)
8. http://www.nilc.icmc.usp.br/lacioweb
9. Kettunen, K.; Kunttu, T; Järvelin, K. To stem or lemmatize a highly inflectional language in a probabilistic IR environment? Journal of Documentation, (2005) 65
10. Korenius, T.; Laurikkala, J.; Järvelin, K.; Juhola, M. Stemming and Lemmatization in the Clustering of Finnish Text Documents. 13th Conference on Information and Knowledge Management, CIKM. Proceedings, (2004) 625-634
11. Krovetz, R. Viewing morphology as an inference process. Artificial Intelligence, (2000) 118:227-294
12. Lapata, M. The Disambiguation of Nominalizations. Computational Linguistics, (2002) 28(3):357-388
13. Mayfield, J.; McNamee, P. Single N-gram Stemming. 26th Annual International ACM SIGIR conference on research and development in IR. Proceedings, (2003) 415-416
14. Orengo, V. M., and C. Huyck. A Stemming Algorithm for the Portuguese Language. 8th Symposium on String Processing and IR, SPIRE. Proceedings, (2001) 186-193.
15. Perini, M. A. Para uma Nova Gramática do Português. São Paulo: Ática (2000)
16. Savary, A., and C. Jacquemin. Reducing Information Variation in Text. In Text- and Speech-triggered Information Access. LNAI, 2705. Springer, (2003) 145-181
17. Sever, H.; Bitirim, Y. FindStem: Analysis and Evaluation of a Turkish Stemming Algorithm. 10th Symposium on String Processing and IR, SPIRE. Proceedings, (2003) 238-251
18. Sparck-Jones, K. ; Walker, S. ; Robertson, S. E. A Probabilistic Model of Information Retrieval: Development and Comparative Experiments – Part 1 and 2. Information Processing and Management, Oxford, (1997) 36(6):779-840
19. Vilares, J.; Barcala, F. M.; Alonso, M. A. Using Syntactic dependency-pairs conflation to improve retrieval performance in Spanish. Computational Linguistics and Intelligent Text Processing. LNCS, 2276. Springer-Verlag , (2002) 381-390
20. Voorhees, E. M. Overview of TREC 2003. NIST Special Publication - SP500-255. 12th Text Retrieval Conference, Gaithersburg (2003)
21. Ziviani, N. Text Operations. In: Baeza-Yates, R.; Ribeiro-Neto, B. Modern Information Retrieval. New York : ACM Press (1999)

A Framework for Integrating Natural Language Tools

João Graça[1], Nuno J. Mamede[1], and João D. Pereira[2]

[1] Spoken Language Systems Lab, L^2F – INESC-ID Lisboa/IST,
Rua Alves Redol 9, 1000-029 Lisboa, Portugal
{joao.graca, nuno.mamede}@l2f.inesc-id.pt

[2] Spoken Language Systems Lab,
Software Eng. Group – INESC-ID Lisboa/IST,
Rua Alves Redol 9, 1000-029 Lisboa, Portugal
joao@inesc-id.id

Abstract. Natural Language processing (NLP) systems are typically characterized by a pipeline architecture in which several independently developed NLP tools, connected as a chain of filters, apply successive transformations to the data that flows through the system. Hence when integrating such tools, one may face problems that lead to information losses, such as: (i) tools discard information from their input which will be required by other tools further along the pipeline; (ii) each tool has its own input/output format.

This work proposes a solution that solves these problems. We offer a framework for NLP systems. The systems built using this framework use a client server architecture, in which the server acts as a blackboard where all tools add/consult data. Data is kept in the server under a conceptual model independent of the client tools, thus allowing the representation of a broad range of linguistic information.

The tools interact with the server through a generic API which allows the creation of new data and the navigation through all the existing data. Moreover, we provide libraries implemented in several programming language that abstract the connection and communication protocol details between the tools and the server, and provide several levels of functionality that simplify server use.

1 Introduction

Natural Language processing (NLP) systems are typically characterized by a pipeline architecture, in which several NLP tools connected as a chain of filters apply successive transformations to the data that flows through the system. Usually, each tool is independently developed by a different person whose focus is on his/her own problem rather than on the future integration of the tool in a broader system. Hence when integrating such tools, several problems arise, which are mainly related to the following: (i) how the tools communicate with each other, (ii) what kind of information flows between the several tools (may cause information lost).

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 110–119, 2006.

At the Spoken Language Systems Lab (L^{2}F), where this work was developed, several NLP systems have already been created. Every time tools were integrated to compose a system most of the detected problems were concerned with the information flow between those tools, which led to the loss of information. These problems are: (i) usually, the output of a tool consists just of the data it has acted upon and it does not contain all the input data. Sometimes this raises a problem if the discarded data is also required by a tool appearing at a later stage of the pipeline; (ii) each tool has its own input/output format so conversions between data formats may be needed when a tool consumes data produced by another one. Moreover, this conversion may not be possible if the descriptive power of each format is distinct; (iii) the formats used by different tools do not establish relations between the input/output data. These relations are useful for aligning information produced at different levels and to avoid the repetition of common data across them.

The proposed solution is a framework using a client server architecture instead of a pipelined architecture. In our solution, the server acts as a blackboard where all NLP tools (clients) add/consult data. The server maintains cross-relations between the existing layers of data. The data is kept in the repository under a conceptual model independent of the client tools. This conceptual model allows the representation of a broad range of linguistic information. The tools interact with the repository through a generic remote API that permits the creation of new data and the navigation through all the existing data. Moreover, this work provides libraries implemented in several programming languages that abstract the connection and communication protocol details between NLP tools and the server, and provide several levels of functionality that simplify the integration of NLP tools.

2 Solution Requirements

We define some requirements that a framework should fulfil in order to solve the problems we detected. These requirements concern the expressive power of the conceptual model and the functionalities offered to the tools. The model must be able to represent linguistic phenomena deemed of interest, and several types of primary data sources (text, speech). The requirements for the conceptual model include:

- Keeping all information produced by an NLP tool on the same layer;
- Representing segmentation ambiguity;
- Representing trees of linguistic elements;
- Representing relational information between linguistic elements;
- Representing classification ambiguity;
- Representing relations between information from different layers (cross-relations).

The requirements defined for the conceptual model were compared against the requirements that are being defined by ISOTC37/SC4 (Terminology and other language resources) to define a standard for linguistic annotation [6]. We found them to be very similar, which strengthened our conviction that any model used to represent linguistic information should follow these requirements.

Finally, we identified the following requirements concerning the functionality that must be supported by the framework:

- It must allow the selection of data based on the identification of each layer;
- It must allow parallel processing of data kept in the server;
- It must guarantee that all data in the repository is kept persistently;
- It must allow interaction with tools written in any programming language.

3 Related Work

We analysed several architectures whose goal was to simplify the creation or integration of NLP tools, towards their usage in NLP systems, namely: the Emdros text database system [9], a text database engine for analysis, and retrieval of analyzed or annotated text; the Natural Language Toolkit [8], a suite of libraries, and programs for symbolic, and statistical natural language processing; the Gate architecture [3], a general architecture for text engineering that promotes the integration of NLP tools by composing them into a pipes and filters architecture; and the Festival speech synthesis system [10], a general framework for building speech synthesis systems.

We also compared some works from the linguistic annotation field, whose focus is on the definition of a logical level for annotation independent of the annotations' physical format. This logical level should be able to represent the most common types of linguistic annotations to promote reuse of annotated corpora. The conceptual model we required to represent the input/output of an NLP tool can be seen as this logical level. In this field we compared two works: the Annotation Graphs Toolkit (AGTK) [7] that is an implementation of the Annotation Graphs formalism [2], the most cited work in this area; and the ATLAS architecture [1], a generalization of the Annotation Graphs formalism to allow the use of multidimensional signals.

The AGTK and the ATLAS architectures do not allow the separation of information into layers. In these architectures, to avoid the loss of information, each tool has to load all previous annotations, and then save them together with its results. This strategy has several drawbacks: first, each tool must know how to manage data which may be unrelated with the tool itself. Second, each tool may have to load and parse extra data upon its initialization and consequently save extra data when terminating. Finally, it is difficult for a tool to handle data from several tools at the same time, because it must merge the common data from the input tools. Moreover, the adoption of the Annotation Graphs model is not possible, mainly because it does not allow the representation of relational information, nor the representation of cross-relations between several data layers. The extensions performed by the ATLAS architecture provide a better representation for conceptually different linguistic phenomena, such as hierarchic trees, and ambiguous segmentations. Furthermore, ATLAS allows the use of every type of data sources. But, even so, this model still presents the same problems as the Annotation Graphs formalism.

The Emdros framework has a representation model that is too restrictive for our objectives. For example, it restricts the media type to text. In Emdros it is not possible to properly represent some types of linguistic phenomena, such as classification ambiguity, or relations between elements.

The NLTK restricts development to the Python programming language, and relies on the Python interpreter to work. Moreover, its underlying conceptual model is not able to fulfill all our requirements, for instance, the representation of ambiguities, such as, classification ambiguity.

The GATE framework presents the same problems as the AGTK formalism, concerning its conceptual model. Moreover, it limits its utilization to the Java programming language. GATE promotes the integration of NLP tools into a pipes and filters architecture, which as we mentioned in the introduction, has some problems that led to the development of this work.

Finally, the Festival framework has a model that does not allow the fulfilment of all our requirements, namely: i) its data source must be text, ii) it cannot represent all types of ambiguities defined in the requirements, iii) it does not allow the concurrent execution of different tools.

4 Proposal

Our proposal consists of a client server architecture. The clients are NLP tools while the server consists of a centralized repository of linguistic information and data sources represented under a conceptual model. Each NLP tool can interact with the server in two ways:

- By using a remote interface independent of the tool's programming language;
- By using a module in its programming language that abstracts the communication and protocol details between the client and the server, and offers an implementation of the conceptual model. The use of the server is simpler using the client library than using the remote API, but the use of the client library requires an implementation of the client library for each programming language used.

4.1 Conceptual Model

The conceptual model is able to represent and relate various types of linguistic information produced by several NLP tools. Besides representing the different linguist phenomena, the model simplifies the use of linguistic information.

The entities composing the conceptual model, described in Fig.1 are:

- *Repository*: a centralized linguistic repository that stores the output of several NLP tools, and organizes that information into layers. Each layer is univocally identified inside the repository. There are two types of layers: *SignalData* and *Analysis*;
- *Data*: data type abstraction used by the *Repository*, e.g. *String*;

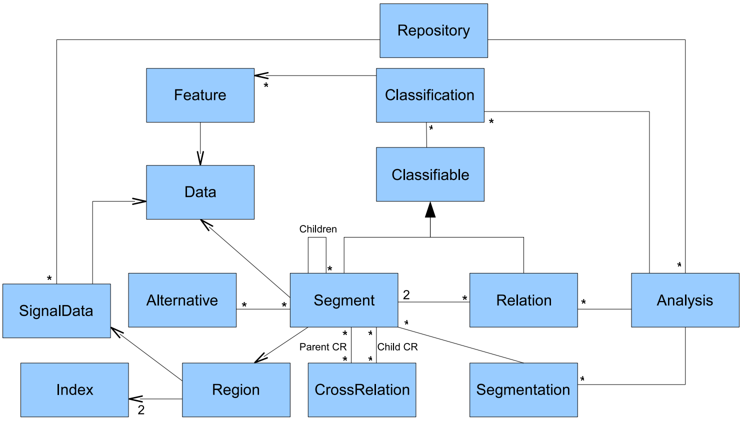


Fig. 1. Conceptual Model class diagram

- *SignalData*: an abstraction of a raw data source, such as a text;
- *Index*: a point in a *SignalData*;
- *Region*: defines a region in a *SignalData*, using a pair of *indexes*;
- *Analysis*: linguistic information other than *SignalData* elements produced by an NLP tool. An *Analysis* may be open or closed indicating whether the tool has already finished the addition of data into the *Repository*. An *Analysis* can only be changed if it is open. The *Analysis* is responsible for the creation of *Segmentations*, *Relations* and *Classifications*;
- *Segment*: a linguistic element, e.g., a word. A *Segment* may contain two Data elements: the original data and the derived data. The original data corresponds to a linguistic element identified in a *SignalData*, while the derived data corresponds to a possible transformation performed over the original data. A *Segment* may be ambiguous, meaning that it has a set of *Alternative* elements for the linguistic element that it identifies. It may be hierarchic, meaning that it has a *Parent Segment* and *Child Segments*. A *Segment* has a set of disjunct *Classification* elements, where each *Classification* assigns a set of characteristics to the *Segment*. It also has a set of Relations, that establish links between two Segments from the same Analysis. A *Segment* may belong to a set of *CrossRelations*, which are used to establish structural relations between *Segments* from different *Analyses*;
- *Segmentation*: a set of sequentially ordered *Segments*, e.g., the words in a sentence. The *Segmentation* is responsible for the creation of *Segments*;
- *Relation*: a link between two segments. For example, the relation between the subject and the verb in a phrase;
- *Classification*: a set of characteristics of a *Segment* or *Relation*. For instance, the morphological features of a word;
- *Alternative*: a set of *Segments* representing different alternatives for an ambiguous linguistic element;
- *Cross-Relation*: a structural relation between *Segments* of distinct *Analyses*.

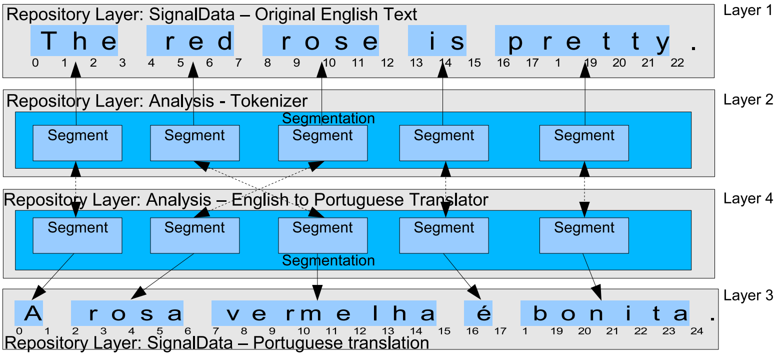


Fig. 2. Data alignment example

Figure 2 shows how *CrossRelations* can be used to align different *SignalData* elements in the translation of the sentence *"The red rose is pretty"* from English to Portuguese, *"A rosa vermelha é bonita"*. The *Repository* contains four layers. The first corresponds to the original *SignalData* containing the English text, where the possible *Indexes* are represented as integers under each character, and the possible *Regions* identifying the words are represented inside a box. The second layer is an *Analysis* containing the segmentation of that text into words: it contains one *Segment* for each word, that uses a *Region* (indicated by an arrow from the *Segment* to the *Region* in the *SignalData*) to refer to the word's text. The *Segmentation* contains those *Segments* in accordance with the word's order in the text. The third layer is a *SignalData* containing the translation of the English text from the first layer, and the fourth layer is an *Analysis* produced by an English to Portuguese translator, which creates a *Segmentation* where each *Segment* represents a Portuguese word from the third layer. The representation of the third and fourth layers is the same as the explained for the first two layers. The alignment between the two texts is achieved by adding *CrossRelations* between *Segments* (dotted arrows) from the corresponding *Analyses*.

4.2 Server Architecture

The server architecture consists of a shared data style. This architecture has the advantages of allowing clients to be added without the server knowledge, and of allowing the integration of the data produced by all the tools. The linguistic information, described using the conceptual model, is managed by the server. The server is organized in three layers: the data layer, the service layer, and the remote interface layer. Each layer can only use the adjacent layers through their interfaces. The layered approach promotes portability, and maintainability since the role of each layer is well identified, and the implementation of each layer can be changed without affecting the other layers.

The Data Layer contains the logic of the application, and uses the conceptual model to represent the linguistic information. Besides representing all the linguistic information, the Data Layer is also responsible for guaranteing the persistence of all linguistic information stored in the server.

The Service Layer [5] defines the server's boundary by providing a set of methods and coordinates the server's response to each method. It is used by the Remote Interface Layer, which handles the specific protocol details, and encapsulates the Data Layer. The Service Layer is responsible for hiding the details regarding the representation of the domain elements of the Data Layer. It transforms the domain elements into Data Transfer Objects (DTO) which will be passed to the client. A DTO [5] is an object with no semantics, and is used to pass information between the client and the server. Each DTO can hold two kinds of information from domain objects: i) identification information used to access domain objects; ii) read-only information from domain objects that may be required by the client.

The Service Layer is also responsible for providing methods that allow the creation of iteration facilities on the client side. Moreover, and since the Service Layer is a single entry point into the server,it is an ideal place to perform logging and authentication actions.

The Service Layer together with the DTOs works as a Remote Facade [5] thus diminishing the number of remote calls required for certain operations.

The Remote Interface Layer provides the methods that are available to client tools according to a selected protocol. It communicates with the Service Layer, and is responsible for serializing the DTOs provided by the Service Layer into their external representation, which will be sent across the connection. It is also responsible for assembling the DTOs back, and pass them to the Services Layer.

4.3 Client Library Architecture

The utilization of the client library allows an NLP tool to abstract from details concerning the communication with the server, and the data exchange protocol. It also provides some high level interfaces that may simplify the integration of NLP tools. The client library uses a layered architecture, each layer is described in the rest of this subsection.

The Client Stub Layer is responsible for communicating with the server under the chosen protocol, through the server's Remote Interface layer. All the other layers of the client library depend and use this layer. This way the other layers are independent of the specific communication protocol that is being used.

The Conceptual Model Layer implements the conceptual model. It allows NLP tools to use the conceptual model as their object model, thus simplifying their creation. Since the concepts used by NLP tools are usually similar, by using the conceptual model we desire to avoid the definition of an equivalent one every time a new tool is created. In addition, by using only the interfaces provided by the Conceptual Model layer its concrete implementation can be changed without changing the tool. This way, an NLP tool can be used as a stand-alone tool or as a client tool connected to the shared repository just by changing the implementation of the Conceptual Model Layer.

The Conceptual Model layer elements are proxies for the elements of the Conceptual Model in the server. The methods performed on those elements are delegated into the corresponding elements in the server.

The repository can be used concurrently by several NLP tools, so it is possible that a tool consumes information that is being produced by another tool at the same time. If the consumer is faster than the producer and depletes the data that is being produced, the consumer may finish its processing due to a lack of data before what was expected. To avoid this situation the iterators on the client side have a blocking behaviour. The method `hasNext()` only returns false when the Analysis that contains the data that is being iterated is closed and there are no more elements to iterate. However, this policy can result in a deadlock to the consumer tool if the producer tool ends abruptly without closing its Analysis. So, we introduced a time limit for which a client method can be blocked in the method `hasNext()`.

The Extra Layers Layer provides extensibility to the client library. It represents new layers that can be added on top of previous ones, enabling the creation of domain specific layers, which may simplify the creation of new NLP tools. For example, a part-of-speech tagger could use a layer that provides the concepts of word, phrase and text, with methods such as `nextWord()`, and `addGender(Word w)`. The use of Extra Layers can also provide semantic meaning to the linguistic information kept in the Repository for a given NLP system.

5 Results and Future work

We implemented our framework in *Java* (around 57 classes) using the XML-RPC protocol provided by the APACHE XML-RPC package. This protocol was chosen because of its simplicity and because it does not impose any restrictions on the programming language used by client tools. Each layer of the server defines a set of interfaces that are used by the other layers. We have implemented a set of specialized classes in the Data Layer to handle text input signals (*TextSignalData*,*StringData*,*TextIndex*). The Service Layer is implemented by two classes: one to implement the methods from its interface, and the other responsible for assembling DTOs from domain objects. The Remote Interface layer is implemented by a class registered in the XML-RPC server as a handler class.

We implemented a client library in *Java* (around 40 classes) using the XML-RPC protocol. The client library Extra Layers layer contains four specific classes to handle text that provide more meaningful methods to NLP tools developers.

We also developed an NLP system to verify the feasibility of our solution. The system is composed of several tools executed sequentially, where each tool uses information produced by previously executed tools. These tools mimic the behaviour of real NLP tools in terms of the input/output data requirements. We have defined the following tools: a text data source creator; a word identification tool, that splits contractions and identifies compound terms, a part-of-speech tagger, a sentence boundary identifier, a syntactic parser, a pos-syntactic parser, and an English-to-Portuguese translator, which keeps both the source text and the translated text aligned.

These tools were implemented using the conceptual model as their object model, avoiding some issues which are usually the developer's responsibility, as the definition of an object model, and the definition of an IO interface. Moreover, each tool uses cross-relations to navigate through the different layers. For example, the translator accesses the original English text segments, and using their cross-relations, accesses their part-of-speech tags disambiguated by the syntactic parser. Using that information it generates the corresponding translation.

This implementation fulfils all the requirements defined, allowing the representation of all types of linguistic phenomena, and complies with the requirements being defined by the ISO committee.

5.1 Future Work

We plan to implement more types of primary data sources, e.g. audio files, and thus enable the use by other NLP systems.

The development of some works in our laboratory, such as, character identification in stories, anaphora resolution, semantic analysis, were suffering from the problems identified in this work. We expect that the use of our framework, that allows them to access all the information produced by NLP tools, and to navigate through related information using cross-relations, simplifies their creation.

Another problem regarding the integration of NLP tools consists in the tags used by each tool to classify linguistic elements. Even if the data structure between two tools is the same, if they use different tag sets to classify the linguistic elements, they will be unable to communicate. This problem was addressed in [4]. Since our framework established a single entry point for the assignment of classifications, a conversion between tagsets could be performed to guarantee that inside the repository all tags were represented in the same way. Each tool would indicate which tag set it requires, and receive the information with the proper tags.

We wish to promote this framework as an annotation framework, and to do so, a graphical interface has to be developed to allow the edition of its data. Moreover, some converter modules have to be defined to allow the use of data annotated in other formalisms, for instance, the *Annotation Graphs* formalism, in which, large corpora have already been annotated, and are publicly available.

As for issues that require future research, our framework does not answer two important questions regarding the integration of NLP tools. First, what data must each tool fetch from the repository. For this problem we intend to integrate the repository with a workflow mechanism and a browser. This will enable the creation of NLP systems, by simply selecting from a browser the tools the system should have. The browser will deal with the selection of the data for each tool. Another question is how does an NLP tool interpret the data kept in the repository. Our conceptual model is capable of representing all linguistic information. However, the same linguistic information may be represented in different ways according to its use. For example, if the phonetic transcription of a text is going to be the target of several NLP tools it should be represented as a new data source. On the contrary, the transcription of each word can be represented as a word's attribute if it is not going to be heavily used. In this work we assume that each tool knows the exact representation of the data. This approach might be too restrictive in terms of extensibility. Some research work should be done in this field, namely in the definition of a meta language, that allows each tool to define its data pre-requirements and pos-requirements, and a way to match this information automatically.

Acknowledgements

This paper was partially supported by project POSI/PLP/41319/2001.

References

1. S. Bird, D. Day, J. Garofolo, J. Henderson, C. Laprun, and M. Liberman. Atlas: A flexible and extensible architecture for linguistic annotation, 2000.
2. Steven Bird and Mark Liberman. A formal framework for linguistic annotation. Technical Report MS-CIS-99-01, Philadelphia, Pennsylvania, 1999.
3. K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Eng.*, 10(3/4):349—373, 2004.
4. David Manuel Martins de Matos. *Construção de Sistemas de Geração Automática de Língua Natural.* PhD thesis, IST - UTL, July 2005.
5. Martin Fowler. *Patterns of Enterprise Application Architecture.* Addison-Wesley Professional, November 2002.
6. N. Ide, L. Romary, and E. de la. International standard for a linguistic annotation framework, 2003.
7. HaeJoong Lee Kazuaki Maeda, Xiaoyi Ma and Steven Bird. *The Annotation Graphs Toolkit (Version 1.0): Application Developer's Manual.* Linguistic Data Consortium, University of Pennsylvania, January 2002.
8. Edward Loper and Steven Bird. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028, 2002.
9. Ulrik Petersen. Emdros - a text database engine for analyzed or annotated text. In *Colling*, 2004.
10. P. Taylor, A. Black, and R. Caley. The architecture of the the festival speech synthesis system, 1998.

Methods and Tools for Encoding the WordNet.Br Sentences, Concept Glosses, and Conceptual-Semantic Relations

Bento C. Dias-da-Silva[1], Ariani Di Felippo[2], and Ricardo Hasegawa[3]

[1,2] Centro de Estudos Lingüísticos e Computacionais da Linguagem – CELiC,
Faculdade de Ciências e Letras – Universidade Estadual Paulista (UNESP),
Caixa Postal 174 – 14.800-901, Araraquara, SP, Brazil
[1,2,3] Núcleo Interinstitucional de Lingüística Computacional – NILC,
Instituto de Ciências Matemáticas e de Computação – Universidade de SãoPaulo (USP),
Caixa Postal 668 – 13.560-970, São Carlos, SP, Brazil
bento@fclar.unesp.br, arianidf@uol.com.br, rh@icmc.usp.br

Abstract. This paper presents the overall methodology that has been used to encode both the Brazilian Portuguese WordNet (WordNet.Br) standard language-independent conceptual-semantic relations (hyponymy, co-hyponymy, meronymy, cause, and entailment) and the so-called cross-lingual conceptual-semantic relations between different wordnets. Accordingly, after contextualizing the project and outlining the current lexical database structure and statistics, it describes the WordNet.Br editing GUI that was designed to aid the linguist in carrying out the tasks of building synsets, selecting sample sentences from corpora, writing synset concept glosses, and encoding both language-independent conceptual-semantic relations and cross-lingual conceptual-semantic relations between WordNet.Br and Princeton WordNet.

1 Introduction

On the one hand, NLP community initiatives to devise methods for developing computational lexicons either from scratch or (semi-)automatically from machine readable dictionaries (MRD) have attested how time-consuming and prone to flaws is to code lexicons for NLP applications [1], [2], [3]. In fact, the bulk of the problem has to do with the amount, the variety, and the complexity of specialized and interrelated information lexicon developers have to cope with and to encode in the database: phonetic/graphemic, morphological, syntactic, semantic, and even illocutionary bits of information [4].

On the other hand, Princeton WordNet (PWN), a successful psycholinguistic experiment, has set the pattern for compiling bulky relational lexicons since its inception in the 1980´s. PWN is basically an on-line relational semantic database combining the design of both a dictionary and a thesaurus. Like a standard dictionary, it covers nouns, verbs, adjectives, and adverbs. After 18 years of research, its 1998 database version (v.1.6) contained about 94,000 nouns, 10,000 verbs, 20,000 adjectives, and 1,500 adverbs [5]. Like a thesaurus, words are grouped in terms of concepts, which are, in turn, represented in terms of synonym sets (*synsets*), i.e. sets of words of the

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 120 – 130, 2006.

same syntactic category that lexicalizes the same concept. Its web structure makes it possible for the user to find a word meaning not only in terms of other words of the same synset but also in terms of its relations to other words in other synsets as well. Despite the fact that PWN is essentially a particular semantic network, its sought-after NLP applications have been discussed by the research community [6], [7].

Structured along the same lines as PWN, wordnets of other languages are under development. The outstanding multilingual initiative is EuroWordNet (EWN) [8], a multilingual database containing monolingual wordnets and equivalence relations for each language synset to the closest concept from the so-called Inter-Lingual-Index (ILI)[1], which enables cross-lingual comparison of words, concept lexicalizations, and meaning relations in different wordnets [9].

Launched in 2003, the WordNet.Br (Brazilian Portuguese WordNet, WBR) extends the Brazilian Portuguese Thesaurus [10], [11]. It is currently being refined, augmented, and upgraded. The improvements include the encoding of the following bits of information in to the database: (a) the co-text sentence for each word-form in a synset; (b) the concept gloss for each synset; and (c) the relevant language-independent hierarchical conceptual-semantic relations of hypernymy[2], hyponymy[3], meronymy (part-whole relation), entailment[4] and cause[5] between synsets.

This paper describes the three aforementioned encoding strategies. Section 2 briefly depicts the current WBR database and its editing GUI (Graphical User Interface), designed to aid the linguist in carrying out the tasks of building synsets, selecting co-text sentences from corpora, and writing synset concept glosses. Section 3 addresses issues of cross-linguistic alignment of wordnets by means of the ILI and desribes the conceptual-semantic alignment strategy adopted to link WBR to PWN. Section 4 outlines the semi-automatic strategy for mapping the PWN verb hyponymy and co-hyponymy relations on to the WBR verb database. Section 5 concludes with some further work.

2 The Current WordNet.Br Lexical Database

After three years of research, the current WBR database presents the following figures: 11,000 verbs (4,000 synsets), 17,000 nouns (8,000 synsets), 15,000 adjectives (6,000 synsets), and 1,000 adverbs (500 synsets), amounting to 44,000 words and 18,500 synsets [12].

Assuming a compromise between Human Language Technology and Linguistics, and based on the Artificial Intelligence notion of Knowledge Representation [13], [14], the project applies a three-domain approach methodology to the development of

[1] The ILI is a list made up of each synset of the PWN with its concept gloss (an informal lexicographic definition of the concept evoked by the synset).

[2] The term Y is a hypernym of the term X if the entity denoted by X is a (kind of) entity denoted byY.

[3] If the term Y is a hypernym of the term X then the term X is a hyponym of Y.

[4] The action A1 denoted by the verb X entails the action A2 denoted by the verb Y if A1 cannot be done unless A2 is, or has been, done.

[5] The action A1 denoted by the verb X causes the action A2 denoted by the verb Y.

the database.[6] This approach claims that the linguistic-related information to be computationally modeled, like a rare metal, must be "mined", "molded", and "assembled" into a computer-tractable system [15]. Accordingly, the process of implementing the database core is developed in the following complementary domains: (a) in *the linguistic-related domain*, the lexical resources (dictionaries and text corpora), the lexical and conceptual-semantic relations, and a kind of natural language ontology of concepts ("Base Concepts" and "Top Ontology" [16]) are mined; (b) in *the representational domain*, the overall information selected and organized in the preceding domain is molded into a computer-tractable representation (the "synsets", the "lexical matrix", and the wordnet "lexical database" itself) [5]; (c) in *the computational domain*, the computer-tractable representations are assembled by means of the WordNet.Br editing GUI.

2.1 The Linguistic-Related Domain

The WBR database core architecture conforms to the two key representations of the PWN [5]: the *synset* and the *lexical matrix*. Synsets are sets of words built on the basis of the notion of "synonymy in context", i.e. word interchangeability in some context [17].[7] The lexical matrix [18] is intended to capture the "many to many" associations between form and meaning, i.e. it associates word forms and the concepts they lexicalize: the lexical matrix is built up by associating each word to the synsets to which it is a member. Thus, a polysemous word will belong to different synsets, for each synset is intended to represent a unique lexicalized concept.
Given the team of three linguists, the unavailability of Brazilian Portuguese MRDs and other computer tractable resources, and a two-year deadline to present large-scale results, the developers, manually, reused, merged, and tuned synonymy and antonymy information registered in five outstanding standard dictionaries of Brazilian Portuguese (BP): [19], [20], [21], [22], and [23, 24].[8] BP texts available in the NILC Corpus[9] and in the web complemented the project reference corpus.

2.2 The Representational Domain

From the logical point of view, the overall structure of the database is made up of two lists: the List of Headwords (LH), the list of words (arranged in alphabetical order), and the List of Synsets (LS), the list of synsets (Fig.1). Each element of a synset (a word form) is necessarily an element of the LH. Each word is specified for its particular Sense Description (SDv) vector. Each SDv is indexed by three pointers: the "syn-

[6] This project was supported in part by contract 552057/01, with funding provided by The National Council for Scientific and Technological Development (CNPq); in part by grant 2003/03623-7 from The State of São Paulo Research Foundation (FAPESP).

[7] Antonymy, on the other hand, is checked either against morphological properties of words or their dictionary lexicographical information.

[8] The dictionaries were chosen for their pervasive use of synonyny and antonymy to define word senses. In a way, this choice dictated the strategy to proceed the work alphabetically, instead of working by semantic fields.

[9] CETENFolha. Corpus de Extractos de Textos Electrónicos NILC/Folha de S. Paulo. http:// www.linguateca.pt/.

onymy pointer", which identifies a particular synset in the LS; the "antonymy pointer", which identifies a particular antonym synset in the LS; and the "sense pointer", which identifies a particular word form sense number in the SDv. Given such an underlying structure, each synset is linked to its concept gloss via the "concept gloss link", and each word is linked to its co-text sentence via the "co-text sentence link".

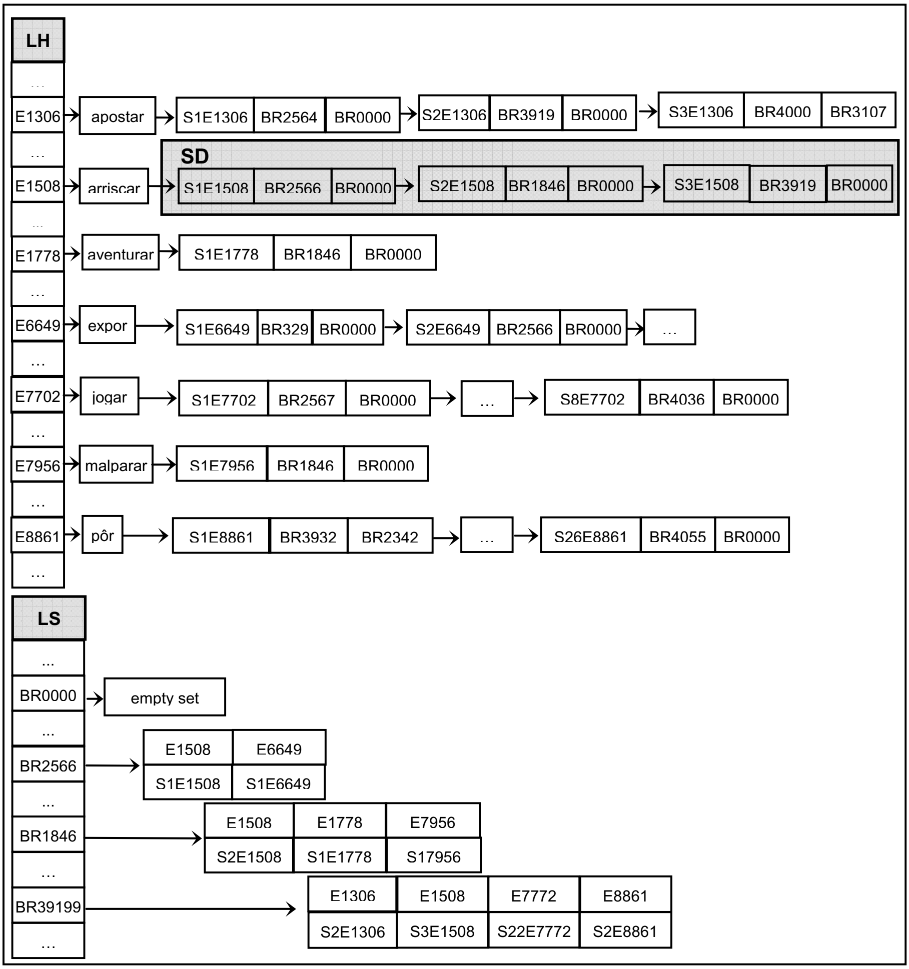
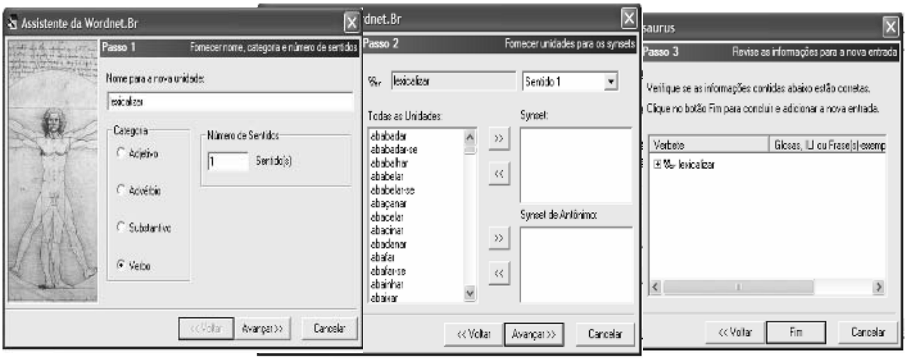


Fig. 1. The WordNet.Br underlying structure

2.3 The Computational Domain

The current WBR editing tool is a Windows®-based GUI. It allows the linguist (a) to create, consult, modify, or save words and synsets; (b) to include co-text sentences for



Nome para a nova unidade: lexicalizar

○ Adjetivo ○ Advérbio ○ Substantivo ● Verbo

1 Sentido(s)

lexicalizar — Sentido 1

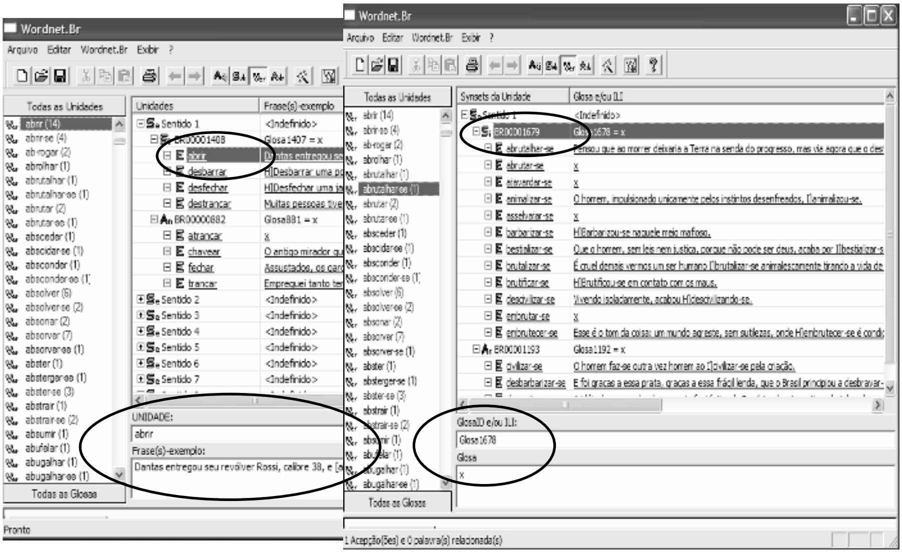
Todas as Unidades: ababadar, ababadar-se, ababalhar, ababelar, ababelar-se, abaçanar, abacelar, abacinar, abadanar, abafar, abafar-se, abainhar, abaixar

Verifique se as informações contidas abaixo estão corretas.

Clique no botão Fim para concluir e adicionar a nova entrada.

lexicalizar

Fig. 2. The procedure for encoding synsets



Unidades	Frase(s)-exemplo
Sentido 1	<Indefinido>
BR00001408	Glosa1407 = x
abrir	Dantas entregou se
desbarrar	H]Desbarrar uma po
desfechar	H]Desfechar uma ja
destrancar	Muitas pessoas tive
BR00000882	Glosa881 = x
atrancar	x
chavear	O antigo mirador qu
fechar	Assustados, os card
trancar	Empreguei tanto ter
Sentido 2	<Indefinido>
Sentido 3	<Indefinido>
Sentido 4	<Indefinido>
Sentido 5	<Indefinido>
Sentido 6	<Indefinido>
Sentido 7	<Indefinido>

UNIDADE: abrir

Frase(s)-exemplo: Dantas entregou seu revólver Rossi, calibre 38, e [a

Synsets da Unidade	Glosa e/ou ILI
Sentido 1	<Indefinido>
BR00001679	Glosa1678 = x
abrutelhar-se	Pensou que ao morrer deixaria a Terra na senda do progresso, mas via agora que o des
abrutar-se	x
alarvar-se	x
animalizar-se	O homem, impulsionado unicamente pelos instintos desenfreados, []animalizou-se.
asselvajar-se	x
barbarizar-se	H]Barbarizou-se naquele meio mafioso.
bestializar-se	Que o homem, sem leis nem justiça, porque não pode ser deus, acaba por []bestializar-s
brutalizar-se	É cruel demais vermos um ser humano []brutalizar-se animalescamente tirando a vida de
brutificar-se	H]Brutificou-se em contato com os maus.
descivilizar-se	Vivendo isoladamente, acabou H]descivilizando-se.
embrutar-se	x
embrutecer-se	Esse é o tom da coisa: um mundo agreste, sem sutilezas, onde H]embrutecer-se é condi
BR00001193	Glosa1192 = x
civilizar-se	O homem faz-se outra vez homem ao []civilizar-se pela criação.
desbarbarizar-se	E foi graças a essa prata, graças a essa fácil lenda, que o Brasil principiou a desbravar-

GlosaID e/ou ILI: Glosa1678

Glosa: x

Fig. 3. The procedure for encoding co-text sentences and concept glosses

each word; (c) to write a concept gloss for each synset; and (d) to generate different types of synset lists (lists arranged by syntactic category, by number of elements, by the degree of homonymy and polysemy, and by co-text sentence) and different statistics. Its main functionalities include the storage and bookkeeping of the general information of the database. The processes of editing (a) words, and (b) co-text sentences and (c) concept glosses can be better understood by an illustrative example. The first GUI dialogue box in Fig. 2 shows the editor at the moment the linguist is constructing synsets that contain the verb "lexicalizar" ("to lexicalize"). In the first dialogue box, the linguist selects the appropriate syntactic category and the expected number of senses (i.e. the number of synsets to be constructed); then, s/he clicks on the “Avançar” button (“Next” button). The second dialogue box “Todas as Unidades” field (“All Unities” field) pops up, i.e. the list of all the words already in the database. To construct a synset (or an antonym synset), the linguist picks out the appropriate words from the list and clicks on the “Avançar” button. In the third dialogue box, s/he concludes the synset construction procedure.

While words and synsets are inserted through dialogue boxes, the co-text sentences and concept glosses are typed in directly in the editor window (Fig.3). The screen shot to the left illustrates the “Frase(s)-exemplo” field (“Co-text sentence” field) when the linguist clicks on a word. The screen shot to the right illustrates the “Glosa” field (“Gloss” field). Similarly, to type in a concept gloss, the linguist clicks on the synset located in the “Todas as Unidades” field.

Currently, the database contains 19,747 co-text sentences selected from the project reference corpus. The following statistics are generated by the editor: Table 1 shows the co-text sentence sources; Table 2 shows the number of co-text sentences per synset.

Table 1. Co-text sentence sources

Source	Number of co-text sentences
NILC Corpus	7,659
Aurélio [19]	732
Houaiss [25]	1,761
Michaelis [20]	858
Web	8,052
unknown	685
Total	19,747

Table 2. Co-text sentence statistics

Number of co-text sentences per synset	Number of synsets
1	18,604
2	521
3	10

3 The Cross-Linguistic Alignment of Wordnets

A rewarding and necessary challenge to the WBR project is to link WBR and PWN (2.0 version) databases. This alignment might permit not only the linguistic investigation of differences and similarities in the lexicalization processes between Brazilian Portuguese and English but also the development of a bilingual lexical database which can be used directly in applications such as cross-language information retrieval involving both languages. Moreover, this bilingual database could generate two types of machine-readable dictionaries: a monolingual Brazilian Portuguese dictionary and a bilingual English-Portuguese dictionary [12]. Furthermore, the possibility of mapping WBR on to PWN might allow the semi-automatic specification of the relevant hierarchical conceptual-semantic relations mentioned in section (1) above.

4 The Alignment Process

The inter-lingual equivalence relations between wordnets are mined in accordance with the types identified by Vossen [8], the so-called, self defining EQ-RELATIONS

(EQ-SYNONYM, EQ-NEAR-SYNONYM, EQ-HAS-HYPERONYM, and EQ-HAS-HYPONYM). Linguistic mismatches (lexical gaps, due largely to cultural gaps, pragmatic differences, and morphological mismatches; over-differentiation or under-differentiation of senses; and fuzzy-matching between synsets) and technical mismatches (mistakes in the choice of inter-lingual equivalence links or in the encoding of language- independent relations across wordnets) as have been described in Peters [9] are also accounted for during the linking procedures. The salient equivalence relations and cross-lingual possible mismatches are molded into a computer-tractable representation that relies on the unstructured list of the PWN synsets, the aforementioned ILI, conceived of as a kind of interlingua used to link different wordnets. Specifically, different wordnets are linked by ILI-records[10]. The ILI-record as a linking device has some technical advantages: (a) it is most beneficial with respect to the effort needed for the development, maintenance, future expansion, and reusability of a multilingual wordnet; (b) it avoids the need to develop and maintain a huge and complex semantic structure to incorporate the meanings encoded by each individual wordnet into the multilingual wordnet; (c) it makes less costly for wordnet developers to add new wordnets to the multingual wordnet [9].

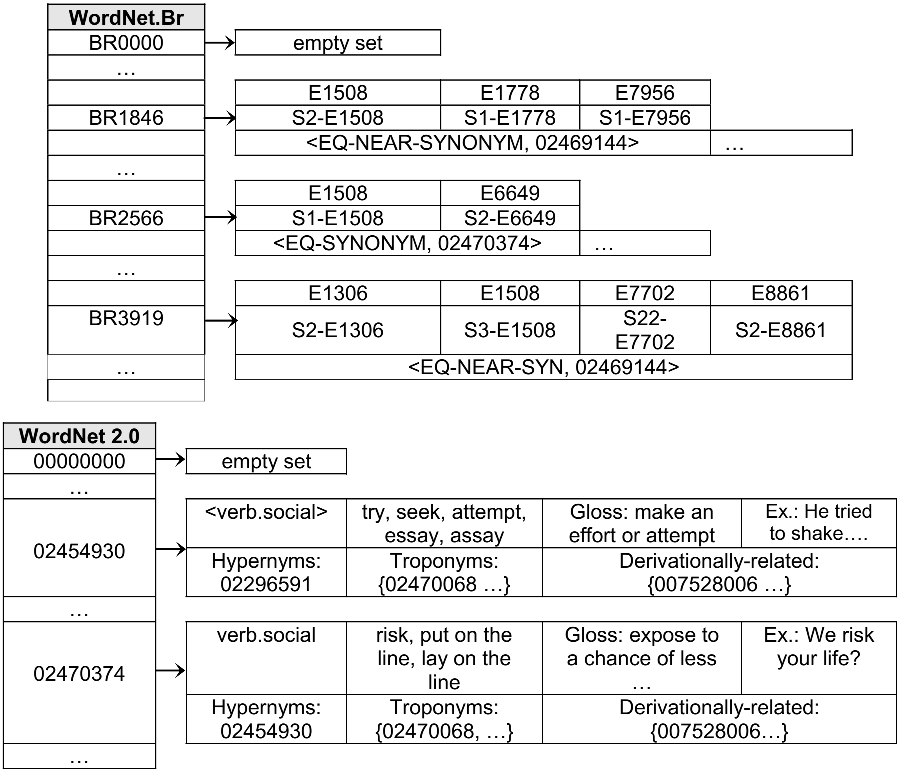
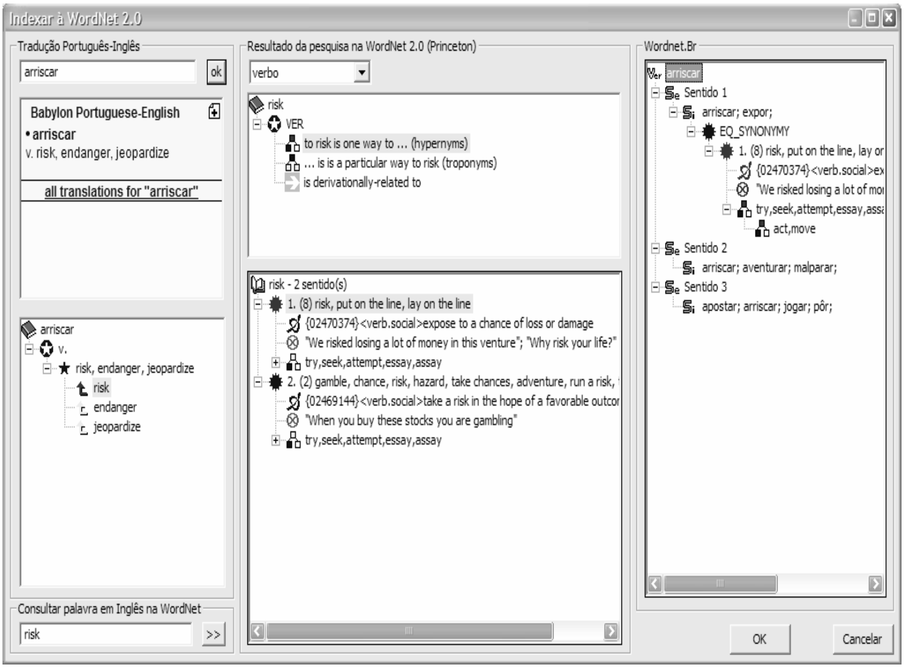


Fig. 4. The synset structure augmented with conceptual-semantic EQ-RELATIONS

[10] An ILI-record is a PWN (version 2.0) synset, its concept gloss and its ID number [9].

To encode the inter-lingual equivalence relations, the overall structure of the database has been further extended as shown in Fig.4. Besides the LH and LS lists and SDv pointers (see 2.2), each synset structure has been augmented with an additional vector to identify both the wordnet standard language-independent concepual-semantic relations (e.g. HYPONYMY, TROPONYMY, CO-HYPONYMY, etc.) and the cross-lingual conceptual-semantic EQ_RELATIONS between synsets of the two wordnets. This new vector enriches the WBR database structure with the following cross-linguistic information: the "universal" synset semantic type (e.g. <verb.social>), the corresponding English synset (e.g. {risk, put on the line, lay on the line}), the English version of the universal concept gloss (e.g. Expose to a chance of loss or damage), the English co-text sentence (e.g. "Why risk your life?"), and EQ-RELATIONS (e.g. EQ-SYNONYM relation).

The current WBR editing GUI has three interconnecting modules. Each module, in turn, makes it possible for the linguist to carry out specific tasks during the procedure for linking synsets across the two wordnets: searching the WBR database, the BP-English dictionary, and the web; searching the PWN database automatically; and linking synsets within WBR and across the two wordnets.


Indexar à WordNet 2.0
Tradução Português-Inglês
ok
Consultar palavra em Inglês na WordNet
>>
Resultado da pesquisa na WordNet 2.0 (Princeton)
Wordnet.Br
OK
Cancelar

arriscar

Babylon Portuguese-English
• arriscar
v. risk, endanger, jeopardize
all translations for "arriscar"

arriscar
v.
risk, endanger, jeopardize
risk
endanger
jeopardize

risk

verbo

risk
VER
to risk is one way to ... (hypernyms)
... is is a particular way to risk (troponyms)
is derivationally-related to

risk - 2 sentido(s)
1. (8) risk, put on the line, lay on the line
{02470374}<verb.social>expose to a chance of loss or damage
"We risked losing a lot of money in this venture"; "Why risk your life?"
try,seek,attempt,essay,assay
2. (2) gamble, chance, risk, hazard, take chances, adventure, run a risk,
{02469144}<verb.social>take a risk in the hope of a favorable outcor
"When you buy these stocks you are gambling"
try,seek,attempt,essay,assay

arriscar
Sentido 1
arriscar; expor;
EQ_SYNONYMY
1. (8) risk, put on the line, lay or
{02470374}<verb.social>ex
"We risked losing a lot of moi
try,seek,attempt,essay,ass
act,move
Sentido 2
arriscar; aventurar; malparar;
Sentido 3
apostar; arriscar; jogar; pôr;

Fig. 5. The three-column WordNet.Br GUI

The linguist starts off the linking procedure by right clicking on a target WBR word. As shown below in Fig. 5, in response to that action the editor displays a three column GUI (the three interconnecting modules), with an online MRD bilingual BP-English dictionary and a web search field at the left, the relevant PWN synsets in the middle, and the WBR synsets that contain the target word to the right. In the first column, (i) the linguist analyzes all possible English words thar are equivalent to the target Brazilian Portuguese word (e.g. the English verbs "risk, endanger, jeopardize" and the BP verb "arriscar"), with recourse to the dictionary and a quick web search;[11] in the middle column, (ii) the linguist analyzes the possible types of equivalence links between the two sets of synsets: the one in the middle column –the sets of synsets of PWN (e.g. the synsets {risk, put on the line, lay on the line} and {gamble, chance, risk, hazard, take chances, adventure, run a risk, take a chance}– and the one in the column to the right –the WBR synsets that contain the targer word (e.g. the synsets {arriscar, expor}, {arriscar, aventurar, malparar}, and {apostar, arriscar, jogar, pôr}).

5 Conclusion

On the way, it is the encoding of (a) a concept gloss for each synset of verbs; (b) a co-text sentence for each verb; (c) the mapping of the WBR verb synsets to its equivalent ILI-records by means of the following equivalence relations EQ-SYNONYM, EQ-NEAR-SYNONYM, EQ-HAS-HYPERONYM, and EQ-HAS-HYPONYM, and the automatic inheritance of PWN´s hipernymy and co-hyponymy relations (See Fig. 6); (d) the conceptual-semantic relations of hypernymy, entailment, and cause between WBR verb synsets.

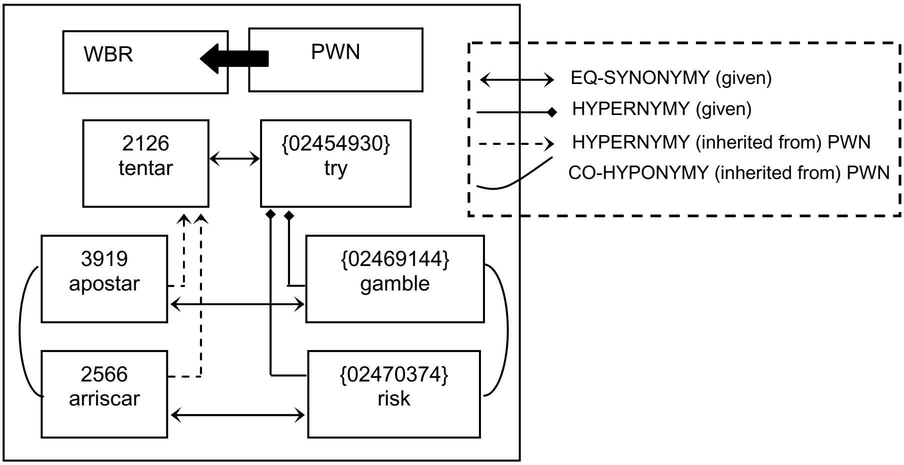


Fig. 6. A sample of an automatic encoding of hypernymy and co-hyponymy

This paper described the overall design and content of the current WBR database, the procedures and tools for encoding synsets, co-text sentences, concept glosses, language-independent conceptual-semantic relations, and conceptual-semantic

[11] It is also possible to select the appropriate English equivalent (e.g. "risk") to trigger the relevant PWN information in the middle column.

equivalence relations between WBR and PWN. It should be stressed that the overall procedures described in this paper are efficient and original if compared to the standard methodologies presented by Rigau et al. [26], which resorts to pre-existing MRD lexical resources.

References

1. Palmer, M. (ed.): Miltilingual resources – Chapter 1. In: Eduard Hovy, Nancy Ide, Robert Frederking, Joseph Mariani, and Antonio Zampolli (eds.): Linguistica Computazionale, Vol. XIV-XV (2001)
2. Hanks, P.: Lexicography. In: The Oxford Handbook of Computational Linguistics, R. Mitkov (ed.), Oxford, Oxford University Press (2003)
3. Di Felippo, A., Pardo, T.A.S., Aluísio, S.M. Proposta de uma metodologia para a identificação dos argumentos dos adjetivos de valência 1 da língua portuguesa a partir de córpus. In: Carderno de Resumos do V Encontro de Corpora, São Carlos, São Paulo (2005) 20-21
4. Handke, J.: The structure of the Lexicon: human versus machine. Berlin: Mouton de Gruyter (1995)
5. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. The MIT Press, Cambridge (1998)
6. Alonge, A., Calzolari, N., Vossen, P., Bloksma, L., Castellon, I., Marti, M.A., Peters, W.: The Linguistic Design of the EuroWordNet Database. Computers and the Humanities, Vol. 32 (1998) 91-115
7. Gonçalo, J., Verdejo, F., Peters, C., Calzolari, N.: Applying EuroWordNet to Cross-Language Text Retrieval. Computers and the Humanities, Vol. 32 (1998) 185-207
8. Vossen, P.: Introduction to EuroWordNet. Computers and the Humanities, Vol. 32(2,3)(1998) 73-89
9. Peters, W., Vossen, P., Díez-Orzas, P., Adriaens, G.: Cross-linguistic Alignment of Wordnets with an Inter-Lingual-Index. Computers and the Humanities, Vol. 32 (1998) 221-251
10. Dias-da-Silva, B.C., Oliveira, M.F.; Moraes, H.R. Groundwork for the development of the Brazilian Portuguese Wordnet. Advances in natural language processing. Berlin: Springer-Verlag (2002)189-196
11. Dias-da-Silva. B.C.; Moraes, H.R. A construção de thesaurus eletrônico para o português do Brasil. Alfa. São Paulo: Editora Unesp, Vol. 47(2) (2003) 101-115
12. Dias-da-Silva, B.C.: Human language technology research and the development of the brazilian portuguese wordnet. In: Proceedings of the 17th International Congress of Linguists – Prague, E. Hajičová, A. Kotěšovcová, J. Mírovský, ed., Matfyzpress, MFF UK (2003) 1-12
13. Hayes-Roth, F.: Expert Systems. In: Encyclopedia of Artificial Intelligence, E. Shapiro (ed.), Wiley, New York (1990) 287-298
14. Durkin, J.: Expert Systems: Design and Development. Prentice Hall International, London (1994)
15. Dias-da-Silva, B. C.: Bridging the Gap Between Linguistic Theory and Natural Language Processing. In: 16th International Congress of Linguists – Paris, B. Caron, ed., Pergamon-Elsevier Science, Oxford (1998) 1-10
16. Rodríguez, H, Climent, S., Vossen, P., Bloksma, L., Peters, W. Alonge, A., Bertagna, F., Roventini, A.: The Top-Down Strategy for Building EuroWordNet: Vocabulary Coverage, Base Concepts and Top-Ontology. Computers and the Humanities, Vol. 32 (1998) 117-152
17. Miller,G.A.:Dictionaries in the Mind. Language and Cognitive Processes,Vol.1(1986)171-185

18. Miller, G.A., Fellbaum, C.: Semantic Networks of English. Cognition 41 (1991) 197-229
19. Ferreira, A. B. H.: Dicionário Aurélio Eletrônico Século XXI. Lexicon, São Paulo, CD-ROM (1999)
20. Weiszflog, W. (ed.): Michaelis Português – Moderno Dicionário da Língua Portuguesa. DTS Software Brasil Ltda, São Paulo, CD-ROM (1998)
21. Barbosa, O.: Grande Dicionário de Sinônimos e Antônimos. Ediouro, Rio de Janeiro, 550 p. (1999)
22. Nascentes, A.: Dicionário de Sinônimos. Nova Fronteira, Rio de Janeiro (1981)
23. Borba, F.S. (coord.): Dicionário Gramatical de Verbos do Português Contemporâneo do Brasil. Editora da Unesp, São Paulo, 600 p. (1990)
24. Borba, F.S.: Dicionário de usos do português do Brasil. São Paulo: Ed. da UNESP (2002)
25. Houaiss, A.: Dicionário Eletrônico Houaiss da Língua Portuguesa. FL Gama Design Ltda., Rio de Janeiro CD-ROM (2001)
26. Rigau, G., Eneko, A.: Semi-automatic methods for WordNet construction. In: 1st Intenational WordNet Conference Tutorial, Mysore, India (2002)

A Multi-agent Approach to Question Answering

Cássia Trojahn dos Santos[1,⋆], Paulo Quaresma[1], Irene Rodrigues[1], and Renata Vieira[2]

[1] Departamento de Informática, Universidade de Évora, Portugal
[2] Pós-Graduação em Computação Aplicada,
Universidade do Vale do Rio dos Sinos, Brazil
{cassia, pq, ipr}@di.uevora.pt, renata@unisinos.br

Abstract. In this paper we present a multi-agent approach to question answering for the Portuguese language. Our proposal is composed by three modules: (1) document and query processing; (2) ontology construction; and (3) answer generation. Each module is composed by multiple cooperative agents which adopt distinct strategies to generate its outputs and cooperate to create a global result. This approach allows the use of different strategies and the reduction of errors introduced by individual methods. The cooperation among the agents aims to reach better solutions in each step of the processing.

1 Introduction

Question answering systems aim to retrieve "answers" to questions rather than full documents or even best-matching passages as most information retrieval systems currently do [6].

Traditional question answering systems employ a single pipeline architecture, consisting roughly of three components: question analysis, search, and answer selection [3]. Typically, each system employs one specific approach in such components. The systems are dependent on the answer search strategy that implement and they can not be able to find a correct answer. This way, there might be others strategies that would be successful at finding an answer.

Recently, the multi-agent approach has received attention from the question answering community. Following this approach, several tasks of query answering process can be distributed between the agents, in order to reach a more easily extensible system. Moreover, the use of several agents encapsulating different strategies in each task can lead the systems to obtain better solutions.

In this paper we present a multi-agent approach to question answering. Our proposal is to extend the architecture of the question answering system for the Portuguese language described in [14]. Such architecture follows a symbolic approach, based on linguistic processing components. This processing includes syntactical analysis of sentences, semantical analysis using discourse representation theory, and semantic/pragmatic interpretation using ontologies and logical inference.

[⋆] Supported by the Programme Alban, the European Union Programme of High Level Scholarships for Latin America, scholarship no. E05D059374BR.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 131–139, 2006.

Our approach is motivated by the success of ensemble methods in machine learning, which have shown great success in improving predictive accuracy. These systems typically employ multiple classifiers to first solve the same problem, then combine the results to provide a final ensemble answer [5]. According to [18] different machine learning techniques applied to the same data set hardly generate the same results – an algorithm A can construct an accurate model for concept X and a weak description for concept Y, while the algorithm B constructs an accurate model for concept Y and a weak model for concept X. Moreover, no algorithm can be the best choice in all possible domains. Each algorithm contains an explicit or implicit bias that leads it to prefer certain generalizations over others [4]. Then, the combination of different learning algorithms can lead to models more accurate. Although determining exactly how to best combine individual results is still an active area of research, a variety of ensemble methods have already been shown to improve predictive performance in various areas of natural language processing [16].

Our multi-agent proposal is composed by three modules: (1) document and query processing; (2) ontology construction; and (3) answer generation. Each module is composed by multiple cooperative agents which adopt distinct strategies to generate its outputs and cooperate to create a global result. This approach allows the use of different strategies and the reduction of errors introduced by the individual ones. Moreover, the cooperation among the agents aims to reach better solutions in each step of the processing.

This paper is structured as follows. Section 2 introduces our multi-agent approach, detailing each agent of the architecture. Section 3 describes some related works regarding multi-agent systems and question answering. Finally, Section 4 comments the final remarks and the future work.

2 The Multi-agent Approach to QA

A QA system should be able to answer queries in natural language, based on information conveyed by a collection of documents. The answer to a specific question is a related set of words and the identification of the document and sentence, which was used as the source of information.

Figure 1 shows the proposed multi-agent architecture. The first module ("document and query processing") is responsible for the document and query processing. The agents of this module act in two steps. First, they extract information from the documents and create a knowledge base. Second, the agents process the query and create the semantic structure of the sentences.

Two types of agents compose the first module: *syntactical analysis agent* and *semantic and pragmatic analysis agent*. The *syntactical analysis agent* is responsible for processing the document or query sentences, generating the syntactic structure of sentences (i.e. syntactical tree, represented in Prolog).

The *semantic and pragmatic analysis agent* transforms the output of the *syntactical analysis agent* in another collection, where each document or query has a semantic representation (i.e, discourse representation structure, DRS [12]). The

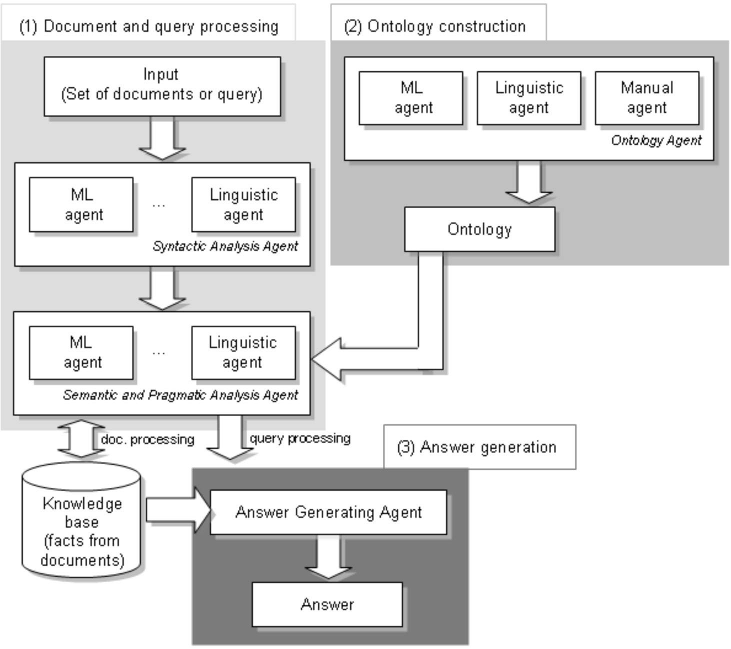


Fig. 1. Proposed multi-agent architecture

ontology is taken into account in this processing. In the phase of document processing, a knowledge base containing instances of the ontology is built as a result.

These agents adopt distinct strategies (i.e, linguistic and learning methods) to generate its results and cooperate to create a final result. This approach is also adopted in the second module of the proposed architecture.

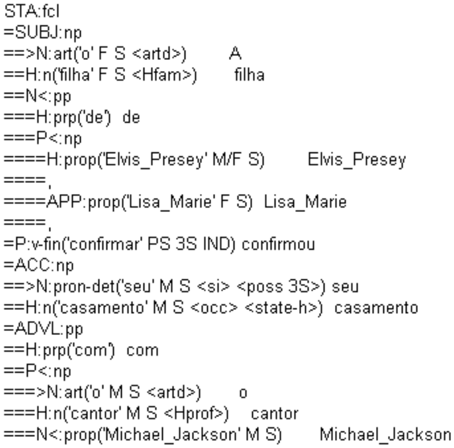
In the second module ("ontology construction") the domain ontology is constructed. Following the same approach adopted in the first module, multiple agents process the documents and cooperate to create an ontology that represents the domain. Each agent is responsible for applying a specific strategy. After, the individual results are combined in a global domain ontology, through cooperation among the agents.

Finally, in the third module ("answer generation"), the answer is generated: a set of words and the identification of the document and sentence where the answer was found. This module is composed by the *answer generating agent* that is responsible for interpreting the query in the knowledge base trough the unification of the discourse entities of the query with documents discourse entities.

In the next sections, each agent is described in more detail.

2.1 Syntactical Analysis Agent

The syntactical agents are responsible for the parsing, adopting machine learning and linguistic methods. According to [8], the problem is to select the most



```
STA:fcl
=SUBJ:np
==>N:art('o' F S <artd>)        A
==H:n('filha' F S <Hfam>)       filha
==N<:pp
===H:prp('de')  de
===P<:np
====H:prop('Elvis_Presey' M/F S)        Elvis_Presey
====,
====APP:prop('Lisa_Marie' F S)  Lisa_Marie
====,
=P:v-fin('confirmar' PS 3S IND) confirmou
=ACC:np
==>N:pron-det('seu' M S <si> <poss 3S>) seu
==H:n('casamento' M S <occ> <state-h>)  casamento
=ADVL:pp
==H:prp('com')  com
==P<:np
===>N:art('o' M S <artd>)       o
===H:n('cantor' M S <Hprof>)    cantor
===N<:prop('Michael_Jackson' M S)       Michael_Jackson
```

Fig. 2. Output of the Palavras parser

plausible syntactic analysis given the great number of analysis a typical parser with a sophisticated grammar may return. For this reason, we are Interested in the use of several approaches for syntactical analysis.

The learning agents model the syntactic structure of sentences using machine learning methods which are capable of learning the structure given correctly annotated documents. Specifically, our agents adopt symbolic (i.e., decision trees [15]) and connexionist (i.e., artificial neural networks [9]) techniques.

Applying these techniques, the agent uses syntactically annotated examples to generate a parser represented by a model induced in the learning phase. This model can be, for instance, in the form of a decision tree from which rules can be extracted. The model can be used to parse previously unseen sentences.

Moreover, the learning agents are based on a shallow parsing approach. Following this approach, rather than produce a detailed syntactic analysis of each sentence, key parts of the syntactic structure are identified or extracted [13]. Such processing include identifying the major phrases in a sentence or extracting the subject, main verb and object from a sentence. According to [8], a full parse often provides more information than needed and sometimes less. In Question Answering, it is interesting the information about specific syntactic-semantic relations such as agent, object, location, time, rather than elaborate configurational syntactic analysis.

In other hand, the linguistic agents use parsers which give morpho-syntactical information of the sentences. For instance, the Palavras [1] parser has a good coverage of the Portuguese language and it has been used by our linguistic agents. As an example of the partial output of this agent, consider the following sentence:

- "A filha de Elvis Presley, Lisa Marie, confirmou seu casamento com o cantor Michael Jackson" (1)
- The daughter of Elvis Presley, Lisa Marie, has confirmed her marriage with the singer Michael Jackson (2)

The syntactical structure of this sentence is presented in Figure 2.

The output of these agents, containing the syntactical structure of the sentences is transformed into a equivalent Prolog representation. Next, the results are combined, generating the final syntactical structure. The advantage of the development of several strategies is the possibility to use these in error detection. According to [13], parsers often make different types of errors and thereby can complement each other.

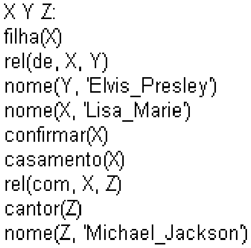
2.2 Semantic and Pragmatic Analysis Agent

Semantic analysis associates a sentence with terms in the ontology. The semantic and pragmatic analysis agents rewrite the syntactical structure in a semantic representation, taking into account the rules obtained from the ontology. For generating these semantic structures, they adopt strategies based on machine learning and linguistic techniques. After generating the individual structures, the agents cooperate to merge their results, creating the final semantic structure. In the document processing phase, the final structure is used to generate a knowledge base containing instances of the ontology. In the query processing, this structure represent the semantic/pragmatic interpretation of the query.

The agents that adopt machine learning techniques try to extract the semantic/pragmatic information (instances of the ontology) taking into account a model induced from learning examples containing the syntactic structures of the sentences. The model is then applied to unseen syntactical structures and the informational sentences corresponding to the rules are extracted.

The linguist agents use a DRT (from discourse representation structure – DRS) to convert the syntactic structure into a semantic structure. Next, they do the pragmatic interpretation using the ontology. As an example of output of this agent, consider the sentence (1). First, the syntactic tree (Figure 2) is rewritten into a DRS. At present, the linguistic agent deals with factual sentences, i.e, sentences with existential quantification over the discourse entities. So, the discourse structures are sets of referents, existentially quantified variables, and sets of conditions, first order predicates.

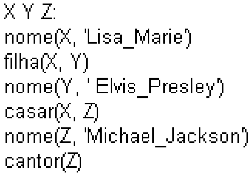
The output of this first processing is shown in Figure 3.



```
X Y Z:
filha(X)
rel(de, X, Y)
nome(Y, 'Elvis_Presley')
nome(X, 'Lisa_Marie')
confirmar(X)
casamento(X)
rel(com, X, Z)
cantor(Z)
nome(Z, 'Michael_Jackson')
```

Fig. 3. Output of the semantic analysis by the linguistic agent

Second, the agent takes as input the semantic information and interprets it using the rules obtained from the ontology and the information in the database.



```
X Y Z:
nome(X, 'Lisa_Marie')
filha(X, Y)
nome(Y, ' Elvis_Presley')
casar(X, Z)
nome(Z, 'Michael_Jackson')
cantor(Z)
```

Fig. 4. Output of the pragmatic analysis by the linguistic agent

In order to obtain a good interpretation, the linguistic agent searches for the best explanation that supports the sentence logical form. This strategy for pragmatic interpretation was initially proposed by [10]. The knowledge base for the pragmatic interpretation is built from the ontology description. The inference in this base uses abduction and finite domain constraint solvers. As a result of this processing, a new DRS is generated (Figure 4).

2.3 Ontology Agent

An ontology is used to model domain knowledge, defining classes and relation between these classes. In our architecture, the ontologies are generated from multiple agents encapsulating strategies based on machine learning, NLP techniques, and manual methods. It is made by merging the results of each agent, generating a global ontology. The objective is to explore the characteristics of different strategies.

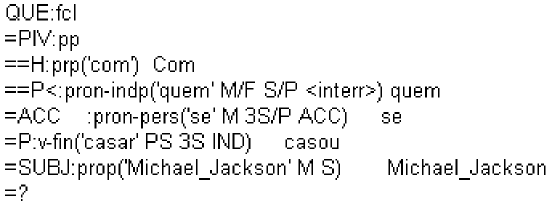
The learning ontology agents are based in the automatic ontology building approach, via machine learning techniques. These agents receive as input annotated documents and learn hierarchical relations between concepts.

The strategies we pretend to use in our NLP agents are based on the approach proposed by [17] which aims to generate ontologies automatically from the document collection. Basically, this approach has the following steps:

- definition of an initial top-level ontology;
- identification of concepts referred in the documents and extraction of its properties;
- identification of relations between the identified concepts;
- creation of an ontology using the identified concepts and relations;
- merge of the created ontology with the initial ontology.

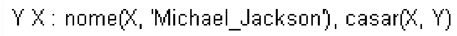
Finally, the manual agents receive as input an ontology manually created and they are responsible for converting the ontology in a standard format.

These three types of agents use OWL (Ontology Web Language) to represent theirs outputs. The global ontology is generated by cooperation among the learning, NLP and manual agents. It is used by the semantic and pragmatic analysis agents in the extraction of facts from the documents and generation of instances of the ontology, which are inserted in the knowledge base (document processing). In the query processing, the ontology is used to interpret the syntactic structure and convert it into a semantic structure. We point out that the global ontology



```
QUE:fcl
=PIV:pp
==H:prp('com')  Com
==P<:pron-indp('quem' M/F S/P <interr>) quem
=ACC    :pron-pers('se' M 3S/P ACC)     se
=P:v-fin('casar' PS 3S IND)     casou
=SUBJ:prop('Michael_Jackson' M S)       Michael_Jackson
=?
```

Fig. 5. Syntactic structure of the question



```
Y X : nome(X, 'Michael_Jackson'), casar(X, Y)
```

Fig. 6. DRS of the question after semantic and pragmatic analysis

is generated for a given domain/collection and, then, used in the processing of all questions related to this domain.

2.4 Answer Generation Agent

The answer generation agent processes the query in two steps:

- identification of the database referent that unifies with the referent of the interrogative pronoun in the question;
- retrieval of the referent proprieties and generation of the answer.

In order to illustrate this process, suppose the following question (3).

- "Com quem casou Michael Jackson?" (3)
- Who Michael Jackson married to? (4)

This question is represented by the following syntactic structure (Figure 5), when it is analyzed by a linguistic agent. The final output of this question, after semantic and pragmatic analysis using the knowledge ontology, is represented in Figure 6.

In order to perform the first step of the answer generation, the agent keeps the referent variables of the question and try to prove the conditions of the semantic structures in the knowledge base. If the conditions can be satisfied in the knowledge, the structures are unified with the identifiers of the individuals.

The next step is to retrieve the words that constitute the answer. In this phase, the agent should retrieve the conditions about the identified referent A and choose which ones better characterize the entity. Our first option is to choose a condition with the predicate *nome*(nome(A,Nome)).

In order to choose the best answer to a question, the agent takes into account the syntactical category of the words that may appear in the answer and it tries to avoid answers with words that appear in the question.

3 Related Work

A multi-agent system, called MASAQ, for answering users' questions based on the knowledge base is presented by [7]. The system consists of four major components: (1) a natural language user interface; (2) an encyclopedic knowledge base covering 21 domains; (3) a communication protocol based XML and KQML; and (4) an executable agent specification language for developing domain-specific multi-agent systems for answering or reasoning about users' questions. Experiments have demonstrated that MASAQ can reason all the knowledge of 21 domains efficiently. An architecture composed by multiple answering agents is proposed by [2]. The agents adopt different processing strategies and consult different knowledge sources in identifying answers to given questions. They employ resolution mechanisms to combine the results produced by the individual answering agents. Experimental results show significant performance improvement over their single-strategy, single-source baselines (35.0% relative improvement in the number of correct answers and 32.8% improvement in average precision).

The proposal of [16] is to evaluate empirically whether combining the outputs of several systems can improve over the performance of any individual system. Seven different natural language algorithms were incorporate in the system (e.g. only a few systems included any semantic processing, and even fewer included coreference). The ensemble experiments showed the utility of majority voting as a method for combining the output of such systems. A similar ensemble approach is presented by [11]. The system combines six radically different QA strategies in the TREC setting. They investigate the impact of various weighted voting techniques (including question type dependent).

Our approach approximates of the work proposed by [7] in sense of the use of agents in the several steps of query and answer processing. However, these authors do not use different strategies (i.e. multiple agents) in each step of the processing as we propose in our work. The interesting in their work is the agent's reasoning about users' questions. We also intend to explore this aspect.

Similarly to [2], we adopt several agents encapsuling different processing strategies. However, we are interested, specifically, in agents using different machine learning and linguist techniques and propose to merge the individual results by cooperation among the agents. Considering the first aspect, our proposal approximates of the work of [16] and [11], which intend to combine individual results, but without the use of cooperation mechanisms.

4 Final Remarks and Future Works

In this paper we presented a multi-agent approach to question answering. Our proposal aims to extend the architecture of the question answering system for the Portuguese language described in [14]. We proposed to use multiple cooperative agents which adopt distinct strategies to generate its outputs. This approach allows to use different approaches and reduce the errors introduced by the individual strategies. The cooperation among the agents aims to reach the better solutions in each step of the processing.

At present, the linguistic agents which are responsible for the syntactical and semantic/pragmatic analysis are implemented. The ontology has been manually created.

As future work, several tasks can be cited: (1) implement the learning agents, defining the techniques which will be used in each step of processing; (2) define and implement the methods to learning ontologies; (3) define and implement the mechanisms of cooperation among the agents; (4) explore the automatically creation of ontologies; and (5) test our system in different domains.

References

1. E. Bick. *The Parsing System Palavras.* Aarhus University Press, 2000.
2. J. Chu-Carroll, K. Czuba, J. M. Prager, and A. Ittycheriah. In question answering, two heads are better than one. In *Proceedings of HLT-NAACL 2003*, 2003.
3. J. Chu-Carroll, J. M. Prager, C. A. Welty, K. Czuba, and D. A. Ferrucci. A multi-strategy and multi-source approach to question answering. In *TREC*, 2002.
4. T. Dietterich. Limitations on inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 124–128, 1989.
5. T. Dietterich. Machine learning research: Four current directions. 4(18), 1997.
6. S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 291–298, 2002.
7. Q. Feng, C. Cao, Y. Sui, Y. Zheng, and Q. Qin. Masaq: A multi-agent system for answering questions based on an encyclopedic knowledge base. In *Declarative Agent Languages and Technologies (DALT)*, 2005.
8. J. Hammerton, M. Osborne, S. Armstrong, and W. Daelemans. Introduction to special issue on machine learning approaches to shallow parsing. In *Journal of Machine Learning Research*, pages 551–558, 2002.
9. S. Haykin. *Redes Neurais Artificiais.* Bookman, 2001.
10. J. Hobbs, M. Stickel, D. Appelt, and P. Martin. Interpretation as abduction. Technical Report 499, Ravenswood, November 1990.
11. V. Jijkoun and M. de Rijke. Answer selection in a multi-stream open domain question answering system. In *Proceedings of European Conference on Information Retrieval*, pages 99–111, 2004.
12. H. Kamp and U. Reyle. From discourse to logic:an introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory. 1993.
13. B. Megyesi. *Data-Driven Syntactic Analysis: Methods and Applications for Swedish.* PhD thesis, University of Kungl Tekniska Hogskolan, 2002.
14. P. Quaresma and I. Rodrigues. A logic programming based approach to the qa@clef05 track. In *Cross Language Evaluation Forum: Working Notes for the CLEF 2005 Workshop*, 2005.
15. J. R. Quinlan. C4.5: Programs for machine learning. 1993.
16. M. Rotaru and D. Litman. Improving question answering for reading comprehension tests by combining multiple systems. In *Proceedings of the AAAI 2005 Workshop on Question Answering in Restricted Domains*, 2005.
17. J. Saias and P. Quaresma. A methodology to create legal ontologies in a logic programming information retrieval system. In *Law and the Semantic Web*, pages 185–200, 2003.
18. H. Viktor and H. Arndt. Combining data mining and human expertise for making decisions, sense and policies. 4(2):33–56, 2000.

Adaptation of Data and Models for Probabilistic Parsing of Portuguese

Benjamin Wing and Jason Baldridge

Department of Linguistics,
University of Texas at Austin,
Austin TX 78712, USA
{benwing, jbaldrid}@mail.utexas.edu

Abstract. We present the first results for recovering word-word dependencies from a probabilistic parser for Portuguese trained on and evaluated against human annotated syntactic analyses. We use the Floresta Sintá(c)tica with the Bikel multi-lingual parsing engine and evaluate performance on both PARSEVAL and unlabeled dependencies. We explore several configurations, both in terms of parameterizing the parser and in terms of enhancements to the trees used for training the parser. Our best configuration achieves 80.6% dependency accuracy on unseen test material, well above adjacency baselines and on par with previous results for unlabeled dependencies.

1 Introduction

Early work on probabilistic parsing focused primarily on English; there is now a growing body of work regarding building treebanks and parsers for other languages. [1] performed one of the first cross-linguistic probabilistic parsing experiments, using the Czech Prague Dependency Treebank [2]. They converted the dependency representations in the treebank to tree structures and then trained various head-driven parsing models [3]. More recent work includes probabilistic parsing for German [4, 5] and French [6].

Portuguese presents many challenges for parsing. Although its nominal inflections are somewhat simpler than languages like Czech and its word order is more restricted, its verbal inflections are significantly more complex. Verbs are conjugated in six person-number combinations and ten synthetic tenses, as well as various non-finite forms. Verbs are lexicalized into three declensional families, and there are numerous subclasses and irregularities. In addition, many verbal endings are identical to inflectional or derivational suffixes used to form nouns, significantly complicating the task of morphological analysis.

A previous statistical parser for *historical* Portuguese, using the Tycho Brahe corpus, was developed by [7]. Using roughly 2000 human-annotated sentences, PARSEVAL f-scores in the 51% to 56% range were obtained with two fairly simple statistical models. A standard Collins parser [8] was implemented by [9, 10] and trained using the CetenFolha corpus (see section 2). However, no manual annotation was then available for this corpus. As a result, the parser

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 140–149, 2006.

was evaluated only qualitatively, on 23 sentences annotated by the author; it is unclear whether these results can be generalized.

There now exists a substantial corpus of Portuguese texts annotated with quasi-dependency structures, the Floresta Sintá(c)tica [11, 12]. Like the corpus used by [10], the analyses are based on the output of the PALAVRAS parser, but for the Floresta, they have been hand-corrected by human annotators to create a gold standard corpus of analyses. However, this resource has until now not been used to train probabilistic parsers for Portuguese.

In this paper, we describe head-driven generative probabilistic parsing models for Portuguese using the Floresta and the Bikel multi-lingual parsing engine [13, 14]. We evaluate parsing performance, using both standard PARSEVAL and unlabeled dependency accuracy, for differing levels of effort in adapting the parser for Portuguese data and adapting the data for the parser. We show that making relatively straightforward changes to the data itself and the parameterization of Bikel's parser – including sensitivity to Portuguese morphology – pays large dividends in performance. Our best model achieves 81.0% unlabeled dependency accuracy and 63.2% PARSEVAL *f*-score on unseen test material. In section 2, we discuss the Floresta and its properties. The next section describes how we produce training material from the Floresta in the appropriate format for the parser and make augmentations to the resulting trees to improve their training utility for the parser. Section 4 introduces the parsing model we use and how we modify it for parsing Portuguese. Section 5 describes how we run our parsing experiments and reports the performance of the various configurations we tested. The last section concludes and describes future work.

2 The Floresta Sintá(c)tica

The Floresta Sintá(c)tica consists of 9,374 sentences and 214,490 tokens, split into two parts of approximately equal size. One part, the CetenFolha (CF), consists of 4,213 sentences and 80,015 tokens taken from the Brazilian newspaper *Folha de São Paulo*. The other, the CetemPúblico (CP), consists of 5,161 sentences and 134,475 tokens taken from the Portuguese newspaper *Público*. The syntactic annotations were produced by hand-correcting the output of the PALAVRAS parser ([15]), a non-statistical parser containing a 75,000-word lexicon and a 2,000-line grammar of inflectional and derivational rules.

Sentence (1) is an example from the corpus. Its analysis is given in Figure 1.

(1) Veículos de resgate estavam a apenas 500 metros do Airbus 300.
Vehicles of rescue were at just 500 meters from.the Airbus 300.
Rescue vehicles were only 500 meters from the Airbus 300.

The annotations provide full morphological analyses of each word and syntactic analyses of each sentence. Each word has a functional tag and part-of-speech tag. `H:n`, for example, specifies a functional tag `H` (head) and a POS tag `n` (noun). `STA:fcl` indicates a finite verbal clause (`fcl`) that is a statement (`STA`). Morphological information for each word is given as the word's lemma and a set

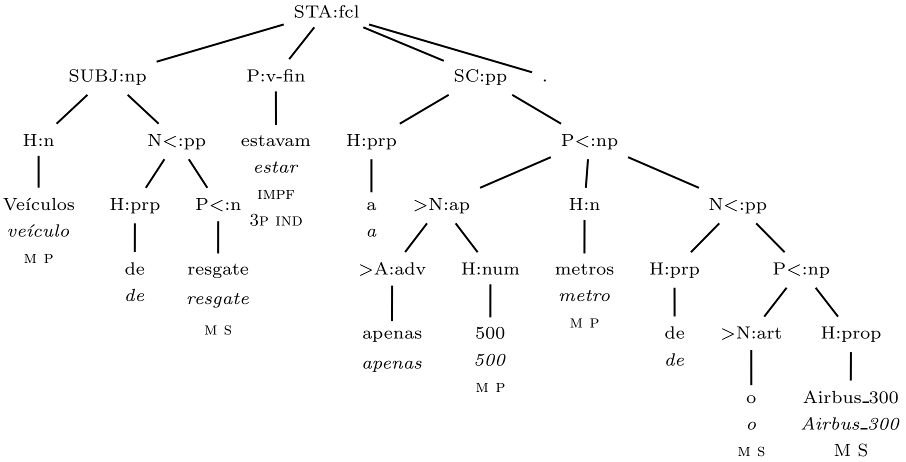


Fig. 1. Floresta tree analysis for sentence (1), from the *arvores deitadas* format

of grammatical features. For example, the annotation “*metro* M P” for *metros* “meters” indicates that it is masculine (M) and plural (P) and its lemma is *metro*.

The tokenization of the text in the syntactic annotations differs quite radically from that of the raw text. The corpus consistently splits combinations of prepositions and determiners, while many named entities and multi-word expressions are joined as one token. In (1), for example, *do* “of the” is split into *de* and *o*, and *Airbus 300* is joined as *Airbus_300*.

The Floresta explicitly indicates heads, arguments and non-argument modifiers. Heads are generally marked with a functional tag of `H`, `P` or `MV`, depending on the constituent type. Arguments are indicated using functional tags such as `SUBJ` (subject), `ACC` (direct object), `PIV` (prepositional object), and `SC` (subject clause). Non-argument modifiers are indicated using functional tags containing a `>` or `<`, indicating a pre-modifier and post-modifier, respectively. The analysis in Figure 1 contains examples of nominal modifiers (`N<` and `>N`), prepositional modifiers (`P<`), and adverbial modifiers (`A<`).

3 Preparing the Training Material

In order to use the Floresta as training material for the Bikel parser, we converted it from its native format into the format used for the Penn Treebank (PTB). We did so in two ways. One is as straightforward as possible and involves no modification to the constituent labels or structures as represented in the *arvores deitadas* (AD) format. The second makes minor changes to the trees and their labels in order to improve their utility for head-driven parsing models.

Certain aspects of the AD format make a complete one-to-one mapping to Penn Treebank format impossible. As a result, we include three transformations to create PTB-style trees. First, punctuation is simply listed as-is in the AD

```
if any children have an AUX functional tag,
  then the leftmost one of these is the head
else if any children have a head marker functional tag (H, P, MV, PMV or PAUX),
  then the leftmost one of these is the head
else if the constituent is a conjunction, then the leftmost conjunct is the head
else if (the label of N is acl (adverbial clause) and
        any children have functional tag COM (complementizer) or PRD (predicator)),
    then the leftmost of these is the head
else if a child has POS tag cu and functional tag ?, then it is the head
else
  collect the set C of children that are neither punctuation nor have a functional
  tag that indicates a non-head
  if C is non-empty
    if N is the root of the tree, then the leftmost of these children is the head
    else choose the head from them in the order:
      clause, conjunction, noun phrase, the leftmost one
  else the leftmost child is the head
```

Fig. 2. Test for determining the head of a node N

format, but requires an associated tag in PTB format. For this, we add tags consistent with Penn Treebank usage; e.g. "`.`", "`?`" and "`!`", are tagged with "`.`". Second, the AD format includes some types of information that cannot be encoded in PTB format, such as morphological analyses and declarations of multiple possible attachment points for some constituents. We simply delete this information as part of the conversion. Finally, the Floresta has discontinuous constituents, which we map into separate constituents.[1]

These simple transformations provide a baseline set of trees that can be used to train a parser. However, it is often the case that steps can be taken to massage the trees in a treebank to improve the parameterization of the parsing models [1, 8, 14]. For the Floresta, we do three main augmentations to the trees: (a) adding explicit head markers, (b) improving the representation of conjunctions, and (c) distinguishing relative clause nodes from other clause level nodes.

Information regarding the heads of constituents in trees is fundamental for deriving dependency relations from treebanks and for parameterizing our parsing models. The PTB format does not mark heads explicitly, so head-driven parsers typically use a complex set of heuristics to determine the head of each constituent. However, heads are (usually) marked explicitly in the Floresta, so we use this to indicate the heads in PTB format by adding `/H` to their label. There are some cases where heads are not marked explicitly – our full test for determining the head of a constituent is given in Figure 2.

To derive the dependency relations, we mostly just create a link from the head word of each non-head child of a constituent to the head word of the constituent's head child. However, to be as consistent as possible with the Portuguese data

[1] There were other minor formatting issues in the conversion, such as standardizing open and closed quotation marks.

in the CoNLL-X shared task on dependency parsing, we need more complex handling of verbal groups (a constituent in the Floresta consisting of a main verb and any corresponding auxiliaries). Verbal subjects (approximated by choosing constituents to the left of a verb that do not have an adverbial tag, i.e. `/ACL`, `/ADV`, `/ADVP` or `/PP`), as well as all punctuation, are dependents of the first auxiliary, but all other constituents are dependents of the main verb. In addition, each verb in the verbal group is dependent on the verb to its left.

Another change we make to tree labels improves the representation of conjunctions. Conjoined clauses in the native Floresta are of type CU, regardless of the type of constituents being conjoined. This causes grammars learned from the treebank to make errors such as conflating noun phrase conjunctions and sentential conjunctions. We thus augment the syntactic type of conjuncts to include the type of the conjoined constituents by using the syntactic type of the head child. This is similar to what was done for Czech by Collins et al. [1].

Following another transformation given in [1], we augment clauses under NPs to distinguish relative clauses from clauses in other circumstances. Essentially, this creates a distinction between a "clause" and a "clause-bar". We identify such clauses by looking for `acl`, `icl` and `fcl` children of `np` constituents.

4 Adapting the Parser for Portuguese

We use Bikel's multi-lingual parsing engine [13, 14] to train and run parsing models for Portuguese. The parser implements and extends the parsing models of Collins [8], which include several lexicalized head-driven generative parsing models that incorporate varying levels of structural information, such as distance features, the complement/adjunct distinction, subcategorization and gaps.

The parsing model we use is essentially Collins' model 2, with the addition of the first-order bigram dependencies described in [1]. With this extension, the generation of a modifier is also dependent on the previous modifier:

$$\prod_{i=1\ldots n+1} \mathcal{P}_l(L_i(l_i)|L_{i-1}, P, h, H)$$

We use Bikel's default approximation of the previous modifier. It is either the (a) START symbol (no previous modifiers), (b) a coordinating conjunction, (c) a punctuation mark, or (d) MISC for all other modifiers.

The Bikel parser allows language-specific extensions to be created. It comes out-of-the-box with support for English, Arabic and Chinese. In addition to using the English package to determine a baseline parsing accuracy, we created a package for Portuguese. This package provides head-finding rules, special handling for when heads are explicitly marked, morphological features, argument/non-argument marking, and some tuning of parser options for the Floresta.

Head-driven parsing models must know the head child of each constituent during training. This information is not encoded in the PTB, so the English package provides a series of head-finding heuristics. For each constituent type, an ordered list of syntactic types is given; the parser searches in turn for a child of each type, assigning the head to the first such child found. For the Portuguese package, we modified these rules as appropriate for the Floresta. We also modified the parser to be aware of the explicit `/H` head indications, as described in Figure 2.[2] When these indications are present, they are marked for every constituent, and thus the head-finding rules are unused. However, the parser will fall back onto these rules as necessary, as in our baseline Portuguese model.

Each language package also can encode features based on morphological properties of a word – these are especially important for unknown words. Five types of features are encoded for each word: capitalization, hyphenation, numeric, inflection, and derivation. The first three indicate, respectively, whether words are capitalized, contain hyphens, or are in the form of numbers. For the most part the code to create them needed no changes. We extensively modified the latter two, however, to handle the morphology of Portuguese.

The inflectional and derivational features indicate the presence of particular suffixes in a word. We created a list of 39 of the recognizably nominal or verbal inflectional endings in Portuguese. This required some care to avoid hitting false positives while at the same time avoiding spreading the features too thin. Thus, we have a single *-rem* to handle the various 3rd plural future subjunctive endings, but separate *-ado* and *-ido* to avoid false positives on nouns like "caldo" and "medo". Furthermore, some endings are not listed at all (e.g. *-o*, *-a*) because they are too ambiguous and are not reliably nominal or verbal. We also modify the handling of plural *-s*; Portuguese plurals nearly always involve a vowel followed by an *-s*, whereas English plurals can have *-s* after various consonants.

We list a series of stop words that should not be segmented. This includes collocations formed by joining multiple words together – these are largely proper names. It also includes words in *-gem* (confusable with verbal *-em*) and a series of common words for the various endings. (E.g. *lugar*, *mar*, *popular* for verbal *-ar*; *classe*, *esse*, *interesse* for verbal *-sse*; *quer*, *qualquer*, *mulher* for verbal *-er*).

We likewise made extensive modifications for the derivational features. We list all common derivational features that are not easily confusable with inflectional features or that rarely occur as inflections. (For example, *-ara* is a literary pluperfect verbal form as well as a nominal ending, but the pluperfect rarely occurs.) We also have special code to handle plurals of suffixes that end in a vowel, without the need to explicitly list each such plural form.

For Collins' model 2, the parser needs to be able to distinguish arguments and non-arguments during training. We found that the heuristic rules used for handling the PTB could be adapted without major work to handle the Floresta as well, since they make explicit reference to the functional tags. Although the PTB, unlike the Floresta, does not explicitly indicate arguments, it does include functional tags of various sorts that are identifiably *not* arguments: these are what are listed in the heuristic rules. We could in principle change how these rules worked for the Floresta, but in practice it worked well to follow the same format

[2] This augmentation is removed at the end of preprocessing to avoid encoding it in the parsing model. This would create difficulties when using tags suggested by a tagger.

and list those functional tags that are clearly not arguments. The remaining nodes are identified as arguments when they occur in the appropriate contexts (e.g., a nominal or clausal child of a clausal constituent). Thus, it sufficed simply to enumerate syntactic tags that identify nominal and clausal constituents and functional tags that cannot be arguments (i.e. modifiers, adverbials and the like).

We made other minor changes to the parser settings. For example, we use Knesser-Ney smoothing instead of the default Witten-Bell, we use an unknown word threshold of two rather than six, and we turn off a number of options that are quite specific to PTB trees. Finally, we restrict the parser so that it makes no unary productions.

5 Experiments

We consider three different parser/data configurations that vary the amount of effort put into adapting the base to the Floresta: BAS-ENG – basic trees with the standard English language package; BAS-PORT – basic trees with the Portuguese package; and AUG-PORT– augmented trees with the Portuguese package. The first represents the laziest approach: do nothing other than ensuring that the trees can be used by the parser. The second makes the parser aware of the language/corpus, while the third involves changing the trees themselves to be more informative to the parser, as described in section 3. We use three different sources of part-of-speech tags: tags obtained from the parser itself (PTAGS), from a tagger[3] (TTAGS), and from the Floresta itself (GTAGS). The latter is used only to show an upperbound on parser performance for each configuration.

We evaluate performance both in terms of standard PARSEVAL f-scores[4] and unlabeled word-word dependencies. We derive our gold standard dependencies as described in section 3. PARSEVAL is a useful way of seeing how well the trees themselves are being modeled by the parser, but the dependency accuracy is the true evaluation. It provides a more clear indication of whether the fundamental relationships recorded in the Floresta are being recovered.

For our experiments, we created a development/training set and test set by randomly sampling from the sentences in the Floresta. The development set has 7497 sentences with 170,527 dependency links, and the test set has 1877 sentences with 42,254 dependency links. We refined our models/configurations using 10-fold cross validation on the development set, and give the performance of our best configuration on the test set.[5]

Figure 3 shows the PARSEVAL f-scores and dependency accuracies for the various configurations. The BAS-ENG configuration unsurprisingly has the worst performance. Though it is not entirely random, we see that simply putting in the relatively minimal effort to create the Portuguese language package leads to large 20-25% absolute improvements in performance in the BAS-PORT configuration. For example, compare PTAGS f-score of 36.3% for BAS-ENG to 60.9% for BAS-

[3] We use the OpenNLP Toolkit maxent tagger, available from `opennlp.sf.net`.

[4] The f-score is calculated as $\frac{2 \times precision \times recall}{precision + recall}$.

[5] We will make the sentence ids in the two sets available to facilitate future comparison.

Model	BAS-ENG			BAS-PORT			AUG-PORT		
	PTAGS	TTAGS	GTAGS	PTAGS	TTAGS	GTAGS	PTAGS	TTAGS	GTAGS
F-score	36.3	37.0	38.6	60.9	60.6	63.8	63.2	63.2	67.1
Dependency Acc.	17.4	18.0	18.4	72.8	73.2	75.7	80.7	81.0	84.0

Fig. 3. PARSEVAL f-scores and dependency accuracy for 10-fold cross-validation experiments with development material

PORT. The transformations to the trees in the AUG-PORT configuration – explicit heads, finer-grained coordination labels, and distinguishing relative clauses – produce a smaller, but significant 2-3% absolute improvement in performance.

The different tagging configurations show that using gold tags results in the best performance. Also, despite the fact that that the tagger tags more accurately than the parser (96.0% vs 94.1%), there is no significant difference in performance between PTAGS and TTAGS for any of the configurations. This is consistent with what was found for Czech [1]. In the TTAGS configuration, we trained the parser on gold standard tags, but tested it with tags from the tagger. Even though the tagger's suggestions are less accurate than the gold standard tags, it can be actually beneficial to use its output in the *training* trees [1]. That way, the tags in training are more like those that the parser will see on tagger-tagged test material. Regardless of how a tagger affects performance, it does have the benefit of speeding up parsing considerably.

The dependency scores in Figure 3 show a similar pattern to the PARSEVAL scores, apart from the BAS-ENG configuration. These scores are compared against left and right linking baselines (i.e., words are dependents of the word to their left or right), which are 26.8% and 22.6%, respectively. The dependency scores for BAS-ENG are worse than either baseline and are relatively much lower than they were for PARSEVAL. This is essentially due to the complete lack of head information, which means that the dependencies extracted from the BAS-ENG parser output are often incorrect because the wrong head was chosen by the parser. PARSEVAL does not reflect this since it only concerns the label and span of a constituent, not the relationships between its children.

The head information provided by the head heuristics in our Portuguese language package is the most likely influence in the considerably better performance of the BAS-PORT configuration, which overwhelmingly improves on both the BAS-ENG configuration and both baselines. When adapting a parser such as Bikel's to a new language, it clearly pays to put in the minimal effort to write even a rough set of reasonably accurate head finding rules.

The even more explicit handling of heads and the tree improvements together then provide a large 8% absolute improvement for AUG-PORT. It is easy to see why the change to coordination labels can make a big difference in the discriminitive capabilities of the parsing model. It makes predictions mostly based on the relationship between children and parent nodes rather than between grandparents and grandchildren. It thus cannot see beyond a simple CU node to know that it contains two NP conjuncts and thereby determine whether they together make a good argument for a verb. The change also prevents coordination of unlike constituents [1]. Distinguishing relative clauses improves the handling of subcategorization of different types of clause level constituents since relative clauses nearly always lack one of the arguments of the verb. Also, they should not be coordinated as a like type with other clauses.

Our best configuration on the development material is AUG-PORT, with no significant difference between using PTAGS or TTAGS. The performance AUG-PORT-PTAGS on the 1877 sentences in the *test* set is an *f*-score of 63.8% (64.7% precision, 62.9% recall) and unlabeled dependency accuracy of 79.9%. For AUG-PORT-TTAGS, we obtain *f*-score of 63.3% (64.1% precision, 62.5% recall) and dependency score of 80.6%. Both dependency scores overwhelmingly beat left and right linking baselines on the test material of 26.9% and 22.6%, respectively, and they are on par with the results obtained for Czech.

We also performed a basic error analysis, investigating the 60 sentences with between 10 to 20 words with the worst dependency figures. The largest source of error was coordination problems (58%), especially in the presence of multiple elements (57% of the coordination problems). The second major source was relativization problems (28%). Some of the additional issues were incorrect handling of subordination (20%), overly eager creation of verb groups (12%), and difficulties handling quoted sentences (12%), fragments (8%), and non-NP subjects (8%). 13% of the sentences revealed errors in the Floresta. This analysis largely vindicates the input transformations we chose. It also points the way towards further work and has suggested some possible solutions – for example, many of the relativization problems may stem from the lack of a clear syntactic category separating relative from non-relative pronouns.

6 Conclusion

In this paper, we provide the first results for probabilistic parsing of modern Portuguese evaluated on significant amounts of human annotated syntactic analyses. We show that an existing probabilistic parser, Bikel's multi-lingual parsing engine, can be readily adapted for Portuguese, and that the accuracy of the parser can be greatly improved with a few relatively straightforward modifications to the parser configuration and to the trees used as training material. Our best configuration on the development material, the AUG-PORT configuration using the tagger tags, achieves 80.6% unlabeled dependency accuracy on unseen test sentences. This result is on par with the accuracy of 80.0% reported for Czech [1].

Much more can be done to improve the parser. In future work, we will perform further modifications to the training trees, such as better handling of discontinuous constituents and introducing finer grained levels of structure instead of the extremely flat trees found in the Floresta. We will also explore lexicalization of models using lemmas as well as full word forms.

Acknowledgements

The authors would like to thank Dan Bikel for advice on modifying and parameterizing the parser. We also thank Susana Afonso, Eckhard Bick, Luis Costa, Diana Santos, and Rui Vilela for their quick and extensive help with the Floresta Sintá(c)tica, and the anonymous reviewers for their feedback.

References

1. Collins, M., Hajic, J., Ramshaw, L., Tillmann, C.: A statistical parser for Czech. In: Proc. of the 37th ACL, College Park, Maryland, USA (1999)
2. Hajic, J.: Building a syntactically annotated corpus: Prague dependency treebank. In: Issues of Valency and Meaning, Karolinum, Prague (1998) 106–132
3. Collins, M.: Three generative, lexicalised models for statistical parsing. In: Proc. of the 35th Annual Meeting of the ACL, Madrid, Spain (1997) 16–23
4. Dubey, A., Keller, F.: Probabilistic parsing for German using sister-head dependencies. In: Proc. of the 41st ACL. (2003) 96–103
5. Dubey, A.: What to do when lexicalization fails: Parsing German with suffix analysis and smoothing. In: Proc. of the 43rd ACL, Ann Arbor, MI (2005) 314–321
6. Arun, A., Keller, F.: Lexicalization in crosslinguistic probabilistic parsing: The case of French. In: Proc. of the 43rd ACL, Ann Arbor, MI, USA (2005) 306–313
7. de Carvalho e Sousa, F.: Analisador sintático estatístico orientado ao núcleo-léxico para a língua portuguesa. Master's thesis, Instituto de Matemática e Estatística da Universidade de São Paulo (2003)
8. Collins, M.: Head-driven statistical models for natural language parsing. Computational Linguistics **29**(4) (2003) 589–638
9. Bonfante, A.G., das Graças Nunes, M.: The implementation process of a statistical parser for Brasilian Portuguese. In: Proc. of the IWPT '01. (2001)
10. Bonfante, A.G.: Parsing Probabilístico para o Português do Brasil. PhD thesis, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (2003)
11. Afonso, S.: Árvores deitadas: Descrição do formato e das opções de análise na Floresta Sintáctica. (2005)
12. Afonso, S., Bick, E., Haber, R., Santos, D.: Floresta sintá(c)tica: A treebank for Portuguese. In Araujo, M.G.R..C.P.S., ed.: Proc. of LREC 2002, Las Palmas de Gran Canaria, Spain (2002) 1698–1703
13. Bikel, D.: Design of a multi-lingual, parallel-processing statistical parsing engine. In: Proc. of the 2nd International Conference on Human Language Technology Research, San Francisco (2002)
14. Bikel, D.: Intricacies of Collins' parsing model. Computational Linguistics **30**(4) (2004) 479–511
15. Bick, E.: The Parsing System PALAVRAS, Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus University Press, Aarhus, Denmark (2000)

A Set of NP-Extraction Rules for Portuguese: Defining, Learning and Pruning

Claudia Oliveira[1], Maria Claudia Freitas[2], Violeta Quental[2],
Cícero Nogueira dos Santos[3], Renato Paes Leme[1], and Lucas Souza[2]

[1] Departamento de Engenharia de Sistemas,
Instituto Militar de Engenharia, Rio de Janeiro, Brazil
[2] Departamento de Letras,
Pontifícia Universidade Católica, Rio de Janeiro, Brazil
[3] Departamento de Informática,
Pontifícia Universidade Católica, Rio de Janeiro, Brazil

Abstract. This paper presents a set of rules for extracting noun phrases from Portuguese texts. We describe how this set was gradually obtained, starting from a machine learned set of transformation rules that was manually reviewed. The noun phrases extracted by these transformations were given as input to another learner that synthesized rules for breaking up complex noun phrases into simpler ones. The results of these processes applied to a Brazilian Portuguese corpus are evaluated.

1 Introduction

Noun phrase (NP) chunking is a fast and robust solution to the identification of NPs in texts. A precise identification of base noun phrases is not only a requirement for information retrieval tasks such as indexing and recalling information, but also seems to be psychologically motivated, since the process of chunking seems to have an important role in human processes of producing, representing and understanding language.

Machine Learning algorithms have been applied in the synthesis of NP models, in a few languages, and the application of these models to the identification tasks has produced very good results in terms of precision and recall. Ramshaw & Marcus [1] reached 92.3% and 91.8%, respectively, for English base NPs. Santos & Oliveira [2] report 86.6% and 85.9%, respectively, in the case of Portuguese NPs, which is a good result considering the complexity of the NP defined in their work.

By definition, a NP chunk, or base NP, is a non-recursive NP [3], or "a noun phrase that does not contain other noun phrase descendants" [4]. In the English literature about base NP chunking (cf. [4], [5]), a base NP includes determiners but does not include any post-modification terms, such as prepositional phrases or clauses. These conditions are too restrictive for the Portuguese NP, typically formed by adding post-modifiers to the head noun, and would reduce, in most cases, the Portuguese chunk to the head noun. Let us consider the following example and its Portuguese translation, in which heads are bolded.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 150–159, 2006.

$_N$[the first Government drug manufacturing **plant**]$_N$
$_N$[a primeira **planta** de $_N$[fabricação de $_N$[remédios do $_N$[governo]$_N$]$_N$]$_N$]$_N$

The obvious translation has necessarily four nested NPs and this structure is a very common construction that has to be extracted as such. Portuguese NPs are, on average, longer and contain more prepositional phrases, giving rise to the problem of prepositional attachment ambiguity.

For the English language, much previous work has been done ([1], [6], [7], [8]) and it seems to be well established that techniques of automatic extracting chunks of information content from unrestricted texts corresponds, in a certain measure, to performing a shallow parsing that will result in base noun phrase chunking.

Brill & Ngai [9] described an experiment in which the task of writing rule lists for base NP annotation was given to a group of students with very little linguistic training. They compared the performance of the students' rules with that of a set of rules automatically learned rules from a corpus, concluding that the human rules came very close to the best machine's performance, in a short amount of time and with a small cost. For base NPs that appeared six or more times in the corpus, the students' recal was 93.5% and the machine's, 93.7%. With NPs that appeared less than five times in the corpus, humans underperform the system. The authors explain this weaker performance by the students' reluctance to take into consideration rules that are not productive. With this result, they propose combining corpus-based knowledge extraction by humans with machine learning techniques.

This paper describes the process of extraction of nominal indexing terms from unrestricted texts. The method is carried out in two phases. In the first phase, a set of NP identification rules is induced from a corpus using Transformation-based Learning, a supervised machine learning algorithm. We report our experiences regarding corpus annotation, testing and evaluation of the NP identification task. In the second phase, a set of NP parsing rules is obtained from a corpus of NPs - a selection of NPs generated as output of the first phase. These rules are designed to break a complex NP into simpler ones. Both sets can be applied to POS-annotated texts.

The remainder of this paper is organized as follows: in Sect. 2 we present the grammatical model of the Portuguese NP we are concerned with; in Sect. 3 we present the overall method, including the first and second phases, and we report our experience in organizing the different instances of the corpus and in dealing with annotation errors; in Sect. 4 we describe our experiments in NP extraction, from the perspective of indexing term generation, and in Sect. 5 we compare our techniques with related approaches; in Sect. 6 we present our concluding remarks.

2 The Grammatical Model of the Portuguese L-NP

In most document retrieval related applications, the target expressions are information rich phrases that will help to infer the contents of the documents. With this principal goal in mind, we define a subset of NPs composed exclusively of lexical noun phrases, that should be the objects of the NP extractor, henceforth the L-NP. The L-NP is the subset of Portuguese NPs characterized by having exactly one nominal head, precluding pronominal NPs and coordinated NPs.

The L-NP model is structured as *head (+complements) (+specifiers)*, where the head is a noun (never a pronoun), a model similar to the lexical SN defined by Radford [10] in the sense that it constitutes a reference independently from other discourse elements. With respect to the elements of the model, we followed the Portuguese NP structure defined in Mateus [11], which can be summarized as follows:

The head is exactly one noun; ordinal numerals (*sexto grau - sixth grade*) are considered as pre-modifiers and cardinal numerals are post-modifiers (*grau seis - grade six*); if the head is elliptical the NP is not considered (*os quatro vieram - the four came*); dates and quantities are not considered;

The complements are adjectival phrases and prepositional phrases; relative clauses and appositives are out;

The specifiers are determiners (articles, possessives, demonstratives and indefinite pronouns) or quantifiers, including numerals as pre-nominal modifiers.

Other considerations covered by the model are:

The continuity of the NP sequence is required, therefore the occurrence of an adverb or other particles, between commas, in the NP will break it (*a elevação, principalmente, da margem esquerda - the elevation, mainly, of the left margin*);

The past participle verb form is always considered as an adjectival specifier except when it occurs with an auxiliary verb (*o barco abandonado vs o barco foi abandonado - the abandoned boat vs the boat was abandoned*) [1].

3 The NP Extraction Task

Our process of extracting NPs from texts uses two sets of rules, both automatically learned. The first set is learned from a POS-annotated full text corpus using Transformation Based Learning (TBL) [6]. The second set is learned from a list of NPs extracted from a full text corpus using the first set of rules. These two learning phases are described in this section.

The training and test corpora used in this study were derived from the Mac-Morpho corpus [12], containing 1.1 million words taken from one year of publication (1994) of the Brazilian newspaper Folha de São Paulo [2]. The corpus is

[1] We are aware that in many instances the past participle verb form introduces a reduced relative clause, but this inherent ambiguity, in Portuguese, is very hard to resolve and it is not distinguishable by the POS tagset of our corpus. For example *Os cartões impressos eram importantes vs Os cartões impressos pela secretária eram importantes - The printed cards were important vs The cards printed by the secretary were important.*

[2] Available at http://www.nilc.icmc.usp.br/lacioweb/, as of October 2005.

annotated with POS tags in the Lacio-Web (LW) Tagset. It was chosen because it consists of Brazilian Portuguese texts of sufficient quantity for the training task, and because the LW tagset was developed to provide a simplified tagset that guarantees such requirements as recoverability, consistency and adaptability to automated learning of POS tagging.

3.1 First Phase: Transformation Based L-NP Rules

The first set of rules was automatically learned from a tagged corpus using Transformation Based Learning (TBL), as illustrated in Fig 1. The pre-processing of the corpus starts by applying the parser PALAVRAS [13] to an unnanotated corpus. The output had to undergo two processes. Firstly, the syntactic annotation had to be interpreted into the tagset {I,O,B} (**I**n NP, **O**ut of NP and **B**order with NP), generating a NP annotated corpus [2]. The second process was the manual revision to enforce the L-NP model, eliminating NPs without a nominal head. Also, some tagging errors were identified and eliminated at this point. The resulted training corpus was used to as input to the TBL learner.

TBL starts with the initial guess classification of an unnanotated training corpus by the baseline system. The resulting classification is compared with the correct one and, whenever a classification error is found, all the rules that can correct it are generated by instantiating the templates with the current token's context. Our templates were manually compiled and empirically refined, as explained in detail in [14]. Normally, a new rule will correct tagging errors, but will also generate some other errors by changing correctly tagged tokens. Therefore, after computing the rules' scores (errors repaired − errors created) the best scoring rule will be selected and stored in order of generation. This rule is applied to the corpus, and the rule generation process will re-start until it fails to produce a rule with a score above an arbitrary threshold. For practical purposes, this minimum score can be tuned to reduce learning time and to avoid overfitting to the training data. The resulting sequence of rules is to be applied in order of generation when annotating a new text.

Testing and Fixing TBL rules. The output of the rule testing process is a corpus annotated with both the original tags and the TBL generated tags. For instance, o_ART_I_I indicates that the token "o_ART" has been annotated in the test corpus as I and by the TBL rules as I. The coincidence between the tags indicate a correct classification. Therefore we are looking for tokens annotated such as primeiro_NUM_I_O and em_PREP_O_I. We classified the errors is eight categories. A description of the categories, examples and the percentage of errors classified in each of them are presented as follows:

1. the prepositional phrase is attached to the verb when it should be attached to the noun - 15.7%;

   (...)manifestou [otimismo quanto às perspectivas de médio prazo da economia] - *showed [optimism toward the mid-term prospects of the economy]* (human)

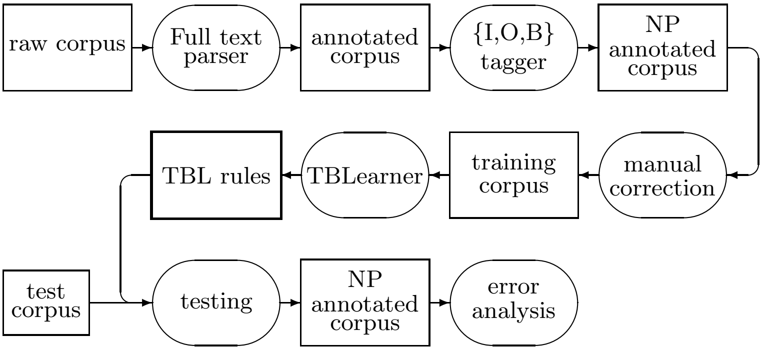


Fig. 1. TBL rule learning phase

(...)manifestou [otimismo quanto às perspectivas de médio prazo] d[a economia] - *showed [optimism toward the mid-term prospects] of [the economy]* (TBL)

2. the prepositional phrase is attached to the noun when it should be attached to the verb - 11%;

   (...)sua habilidade de evitar [impostos no longo prazo] - *their ability to avoid [taxes over long periods of time].* (human)

   (...)sua habilidade de evitar [impostos] n[o longo prazo] - *their ability to avoid [taxes] over [long periods of time]* (TBL)
3. head modifier coordination is not recognized - 5%;

   Os analistas identificaram [as áreas lucrativas e não lucrativas do negócio] - *The analysts identified [the profitable and non-profitable areas of the business]* (human)

   Os analistas identificaram [as áreas lucrativas] e [não lucrativas do negócio] - *The analysts identified [the profitable] and [non-profitable areas of the business]* (TBL)
4. the coordination of two different verb complements are not recognized - 0.7%;

   (...) depende de [justiça social] e de [as autoridades locais] passarem a respeitar a lei - it depends on [social justice] and on [local authorities] respecting the law (human)

   (...) depende de [justiça social e de as autoridades locais] passarem a respeitar a lei - it depends on [social justice and on local authorities] respecting the law (TBL)
5. two different verb complements are merged - 2.5%;

   Suicidou-se com [um tiro] [dois anos depois da morte do irmão] - He committed suicide [with a gun] [two years after his father's death] (human)

   Suicidou-se com [um tiro dois anos depois da morte do irmão] - He committed suicide [with a gun two years after his father's death] (TBL)
6. inconsistencies in the test corpus with respect to quotation marks - 5%;
7. errors derived from incorrect POS tagging in the training corpus - 3%;

8. errors where the TBL rules' annotation was correct and the test corpus annotation was wrong; these errors are derived from both the initial PALAVRAS annotation and our manual corrections following the definition of the L-NP - 41%;
9. others - 17%.

From these numbers we can conclude that 41% of the errors are not generated by the rules but, rather, they were in the test corpus. The proportion of prepositional phrase attachment errors in the legitimate errors is quite high - around 45%.

This error analysis indicates some ways in which the TBL templates can be improved. The most obvious one is lexicalizing some templates to make sure that certain verbs and certain nouns are singled out in terms of their complements. For instance, in our L-NP structure it is undesirable to allow two adjacent nouns in the same NP. It will entail errors such as

> (...) participam d[a banda artistas de novela]" - *integrating [the band artists of soap operas]* (TBL)
> (...) participam d[a banda] [artistas de novela]" - *integrating [the band] [artists of soap operas]* (human)

On the other hand, if the N+N sequence is excluded, a significant number of NPs will be overlooked, such as

> (...) [a palavra justiça] é problemática" - *[the word justice] is problematic* (human)

The solution is to lexicalize the term "palavra" in the TBL templates. This is feasible because the bulk of the instances of the sequence N+N are relatively predictable in Portuguese, e.g., "sangue tipo A" - *blood type A*; or "casa número 20" - *house number 20.*

3.2 Second Phase: L-NP Parsing Rules

The second set of rules is learned from the set of L-NPs extracted from the test corpus. The TBL rules were applied to the POS-tagged test corpus, generating a list of 17,651 L-NPs. From this list, a list of L-NP samples was selected, containing one of each NP-structure found in the full NP list. For instance, 3,149 NPs with the structure (ART N) were found, but only one was selected into the one-each corpus.

The objective of this phase is to identify a set of rules with which we could parse the recursive structure of NPs in Portuguese. As illustrated inj 2, the process starts with the extraction of the sequences of POS that constitute the NPs from the samples list and sorting these sequences by number of POS tags. Let $\mathcal{R}$ be a set of rules that parse NPs. In a simple language, with only nouns and adjectives and where the adjectives are always pre-modifiers ($ADJ^*\ N$), we would have:

$$\mathcal{R} = \{N, ADJ\ NP\}$$

Our learning algorithm proceeds as follows. At first, $\mathcal{R}$ is empty. The list of POS sequences is scanned from the top (NPs with just one component) to the bottom (the largest NPs) and the set $\mathcal{R}$ is gradually built with the following algorithm. For each $NP = [POS_1, POS_2, ..., POS_n]$ in the samples list:

```
$NP = [POS_1\ POS_2\ ...\ POS_n]$
While exists a substring $[POS_i\ ...\ POS_j] \in \mathcal{R}$ from NP
    $NP = [POS_1\ ...\ POS_{i-1}\ NP\ POS_{j+1}\ ...\ POS_n]$
If $NP \neq [NP]$
    Insert $NP$ in $\mathcal{R}$
```

In a nutshell, the algorithm proceeds from the unit NPs, iteratively reducing the complex NPs in terms of the simpler ones. For instance, if at some point in the process $N \in \mathcal{R}$, the following reduction can be performed

$$[ADJ, N] \longrightarrow [ADJ, NP]$$

In order to validate these rules, we applied them to the NPs in the one-each corpus, generating all the sub-NPs derived from them. It was observed that different parsings were possible for the same NP. For example, in the sentence: "Terceira maior siderúrgica da América Latina" - *third largest steelworks in Latin America*, three different bracketings were produced:

[[terceira [maior [siderúrgica]]] de [a [América Latina]]]
[terceira [[maior [siderúrgica]] de [a [América Latina]]]]
[terceira [maior [[siderúrgica] de [a [América Latina]]]]]

Although the correct parsing of a given phrase might be subject to different grammatical approaches, we stay clear of these issues because we evaluate the correctness of the sub-NPs in the brackets, not the parsing tree. In the above example, for each bracketing produced by the rules, we listed the parsed sub-NPs, eliminating the repetitions: América Latina, a América Latina, maior siderúrgica, maior siderúrgica da América Latina, siderúrgica, siderúrgica da América Latina, terceira maior siderúrgica, terceira maior siderúrgica da América Latina.

4 From L-NPs to Terms - Experimental Results

The final set of L-NP parsing rules, after some minor human interventions, is shown in table 1. We lexicalized two rules: in NP([art,pden,NP]), pden (denotative word) was replaced by the instances "então" and "só"; in NP([art,prep,NP]), prep (preposition) was replaced by "cerca de". These rules are marked with a dot in table 1.

The parsing rules were applied to a corpus of 1,343 complex noun phrases; the list of parsed sub-NPs, without repetitions, contained 9,775 items. The experimental results are summarized in table 2.

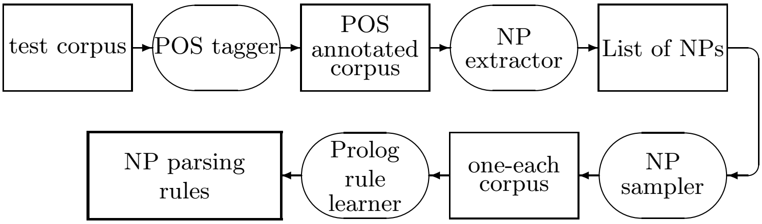


Fig. 2. NP parsing rules learning

Table 1. NP parsing rules

NP([n])
NP([art,NP])
NP([NP,num])
NP([num,NP])
• NP([art,"só",NP])
NP([NP,prep,num])
• NP([art,"cerca de",NP])
NP([NP,prep,art,num])
NP([adv,NP])
NP([NP,prep,art,adj])
NP([nprop])
NP([NP,adj])
NP([NP,pcp])
NP([proadj,NP])
NP([NP,prep,adj])
NP([art,adv,NP])
NP([NP,kc,adj])
NP([NP,prep,prep,NP])
NP([art,adj,kc,NP])
NP([NP,kc,adv,adj])
NP([adj,NP])
NP([NP,adv])
NP([NP,proadj])
• NP([art,"então",NP])
NP([NP,prep,NP])
NP([NP,prep,adv])
NP([NP,kc,num])
NP([proadj,adv,NP])
NP([NP,pden,NP])
NP([NP,prep,proadj,kc,NP])

Table 2. Experimental results

Complex NPs	Parsed NPs	Sub-NPs identified	Sub-NPs correctly identified
1,343	996 (74.16%)	9,775	9,316 (95.30%)

A brief look at the errors is enough to conclude that adjustments in the set of rules could improve the performance a great deal. Nevertheless, certain types of mistake are really tough to eliminate, as they are related to prepositional and adjectival phrase attachments. For instance, "a ação da ONU no Zaire" - *the action of the UN in Zaire* - will yield "a ONU no Zaire", when it should yield "a ação da ONU" and "a ação no Zaire". This last NP would only appear if we were considering discontinuous sub-NPs, but this is not the case.

5 Related Approaches to NP Extraction

We singled out two research papers which share the same objectives as our work and report the use of related methods.

Evans & Zhai [8] describe a hybrid approach to the extraction of meaningful sub-NPs from complex NPs, using both statistics and linguistic heuristics. They go on to show how indexing a text by these NPs improves document retrieval precision and recall. They identify four kinds of phrases that can be indexing terms: in the example "the quality of surface of treated stainless steel strip" the substring "stainless steel" is a nominal multi-word expression; "steel strip" is a head modifier pair; "stainless steel strip" is a sub-compound; and "surface quality" is a cross-preposition modification pair.

Cardie & Pierce [7] explore the role of lexicalization and pruning of grammars for base NP identification. Their framework, the treebank approach, works from a base NP-annotated corpus, extracting rules as sequences of POS tags that constitute a base NP. These rules are pruned in order to increase performance and the rule set thus obtained is evaluated in terms of precision and recall of NPs in the test corpus. The advantages of the lexicalization of rules is also evaluated. Cardie & Pierce's approach is similar to the technique we used to obtain the NP parsing rules.

The fact that we do not have a linguistic resource such as the Penn Treebank[3] for Brazilian Portuguese determines our two phase approach, which is the main difference between our work and Cardie & Pierce's. On the other hand, we want to produce indexing terms from complex NPs, an aim more in line with the work of Evans & Zhai's.

6 Concluding Remarks

We have presented an approach to the NP analysis of unrestricted texts, in order to generate indexing terms for Information Retrieval Systems. As far as we know, there is no other proposal with similar goals and techniques for the Portuguese language.

The necessity of organizing the process in two phases was due to the complexity of the Portuguese NP and to the difficulties related to the availability of annotated corpora and other linguistic resources. These difficulties were also an obstacle to the evaluation of the method in terms of the traditional values of precision and recall.

The simplicity of the rules and speed of their application to a POS-annotated corpus more then compensate for the errors produced at the end of the overall process. That is not to say that we are not concerned about these errors. Some common types are being addressed at the moment. It seems that the L-NP parsing rules tend to over-generate sub-NPs, therefore we were more concerned with computing the precision of the rules rather then their recall.

It would also be useful to include, at the end of the process, a set of post-processing routines, in order to eliminate from the final list NPs which are formal variations of each other, such as the structure (ART N) and (N).

We observed that the most significant errors produced in the two phases can be eliminated with the aid of a lexical resource containing information related to nominal argument structures and multi-word expressions. This is specially true in the case of NP parsing and prepositional phrase attachment errors.

[3] The Penn Treebank Project, at www.cis.upenn.edu/˜ treebank.

References

1. Ramshaw, L., Marcus, M.: Text chunking using transformation-based learning. In Yarovsky, D., Church, K., eds.: Proceedings of the Third Workshop on Very Large Corpora, New Jersey, USA, Association for Computational Linguistics (1995) 82–94
2. dos Santos, C.N., Oliveira, C.: Aplicação de aprendizado baseado em transformações na identificação de sintagmas nominais. In: Anais do XXV Congresso da Sociedade Brasileira de Compútação, Brazil (2005)
3. Abney, S.: Parsing by chunk. In Berwick, R., Abney, S., Tenny, C., eds.: Principle-Based Parsing. Kluwer Academic Publishers (1991)
4. Cardie, C., Pierce, D.: Error-driven pruning of treebank grammars for base noun-phrase identification. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, Ithaca-NY (1998) 218–224
5. Cardie, C., Pierce, D.: The role of lexicalization and pruning for base noun phrase grammars. In: Proceedings of the 16th National Conference on Artificial Intelligence. (1999)
6. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics **21** (1995) 543–565
7. Cardie, C., Wagstaff, K.: Noun phrase coreference as clustering. In: Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, Ithaca-NY (1999) 82–89
8. Evans, D., Zhai, C.: Noun-phrase analysis in unrestricted text for information retrieval. In: Proceedings of the 34th annual meeting on Association for Computational Linguistics. (1996) 17–24
9. Brill, E., Ngai, G.: Man vs. machine: a case study in base noun phrase learning. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. (1999) 65–72
10. Radford, A.: Transformational Syntax. Cambridge University Press (1981)
11. Mateus, M., Brito, A., Duarte, I., Faria, I.: Gramática da língua portuguesa. 4th edn. Ed. Caminho, Lisboa (1994)
12. Marchi, A.R.: Projeto lacio-web: Desafios na construção de um corpus de 1,1 milhão de palavras de textos jornalísticos em português do brasil. In: 51^o Seminário do Grupo de Estudos Lingüísticos do Estado de São Paulo, São Paulo, Brasil (2003)
13. Bick, E.: The Parsing System Palavras: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. PhD thesis, Aarhus University (2000)
14. dos Santos, C.N.: Aprendizado de máquina na identificação dos sintagmas nominais: o caso do português brasileiro. Master's thesis, Instituto Militar de Engenharia, Rio de Janeiro, RJ (2006)

Resolving Portuguese Nominal Anaphora

Jorge C.B. Coelho[1], Vinicius M. Muller[1], Sandra Collovini[1],
Renata Vieira[1], and Lucia H. M. Rino[2]

[1] Universidade do Vale do Rio dos Sinos, Brazil
[2] Universidade Federal de São Carlos, Brazil
{cesarc, viniciusm, sandrac, renatav}@unisinos.br, lucia@dc.ufscar.br

Abstract. This paper presents two approaches to Portuguese anaphor resolution, one based on a morpho-syntactic information and other on semantic features. We present a corpus based evaluation, focusing especially on anaphoric definite descriptions, which includes direct, indirect and associative anaphors.

1 Introduction

This paper presents two approaches to Portuguese anaphora resolution (AR) which are based on morpho-syntactic and semantic layers (head-matching and semantic tags approach, respectively). We focus on nominal anaphors, limiting our research to definite descriptions (DDs) - noun phrases with a definite article (*o, a, os* and *as*).

AR is a very important problem for several tasks in Natural Language Processing, such as Information Extraction, Text Summarization (TS), Machine Translation and others. Dealing with DDs is of utmost importance for TS, our chosen application to contextualize anaphora resolution in this paper. It poses a difficult problem when a text unit that embeds an anaphor is chosen to compose a summary and its antecedent is not. Coherence, in this case, is often damaged as well as the degree and precision of informativity.

This paper is organized as follows: Section 2 details the types of anaphora considered in our work; Section 3 describes the corpus of our study; Section 4 presents our AR approaches; Section 5 discusses related work and refers to the relevance of AR to TS; the last Section presents the conclusions.

2 Anaphoric Definite Descriptions

In an anaphoric relation, the complete interpretation of an expression, the anaphor, is dependent on a previous expression of the discourse, the antecedent. In this work we consider distinct anaphoric relations, concerning DDs, as exemplified (DDs are presented in boldface and antecedents are underlined):

New DDs: definite referring expressions that introduce new entities in the discourse. We consider two types of New DDs, *Brand-new* and *Associative anaphors.*

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 160–169, 2006.

- **Brand-New.** (discourse-new or non-anaphoric)[1]: DDs that introduce entities which are new in the discourse:
  1. ***A Folha de São Paulo.*** *apresentou as listas apreendidas na operação contra o crime organizado.*
  The Folha de São Paulo. *presented the lists arrested in the operation against the organized crime.*

- **Associative anaphors:** DDs that refer to entities that need a semantic connection with an antecedent for their interpretation (the semantic relation is different from identity):
  2. *A Folha de São Paulo apresentou as listas apreendidas na operação contra o crime organizado. O jornal tentou ouvir* ***o delegado encarregado****.*
  The Folha de São Paulo presented the lists arrested in the operation against the organized crime. The newspaper tried to listen to ***the police chief in charge****.*

Old DDs: refer to entities mentioned in the previous discourse. They can be either *Indirect* or *Direct anaphors.*

- **Indirect anaphors**: DDs that have an identity relation with their antecedents through different head-nouns:
  3. *A Folha de São Paulo apresentou as listas apreendidas ...* ***O jornal*** *tentou ouvir ....*
  The Folha de São Paulo presented the lists arrested ... ***The newspaper*** *tried to listen ...*

- **Direct anaphors**: DDs that have an identity relation with their antecedents through the same head-noun:
  4. *...as listas apreendidas na operação contra o crime organizado. Alguns delegados também são citados n****as listas****.*
  ... the lists arrested in the operation against the organized crime*. Some police chiefs are also mentioned in* ***the lists****.*

Based on these classes we developed a corpus study which is presented next.

3 Corpus Study

Our corpus amounts to 20 newspaper articles from Folha de São Paulo, written in Brazilian Portuguese and extracted from NILC corpus[2]. The corpus was automatically annotated with linguistic information using the parser PALAVRAS[3] [1], and manually annotated for anaphoricity using the MMAX tool [5]. The following steps were sequentially accomplished manually: distinguishing New and Old DDs; pointing to the antecedent; and discriminating *New*

[1] In brackets we present other nomenclature used often in the related literature.
[2] http://www.nilc.icmp.usp.br/nilc
[3] http://visl.sdu.dk/visl/pt/parsing/automatic

Table 1. Manual Annotation

Classes	**# of DDs**	**% of Total**
Associative anaphora	21	8%
Indirect anaphora	62	24%
Direct anaphora	179	68%
Total	**262**	**100%**

Table 2. Semantic relations - Associative and Indirect anaphora

Associative anaphora			**Indirect anaphora**		
Relation	**#**	**% of Total**	**Relation**	**#**	**% of Total**
Meronymy	19	90%	Synonymy	39	63%
Hypernymy	1	5%	Hypernymy	11	18%
Other semantic relations	1	5%	Other semantic relations	12	19%
Total	**21**	**100%**	**Total**	**62**	**100%**

and *Old* according to their subclasses. Four subjects annotated the corpus, identifying 1605 noun phrases and 262 cases of anaphoric DDs and agreeing upon them as follows: all agreed upon 73% cases, three agreed on 22% , and only two annotators agreed on 5% cases The resulting distribution is presented in Table 1.

In *Associative* and *Indirect anaphors*, semantic relations with their antecedents play an essential role for resolution. We analyzed each case according to relations of synonymy, meronymy, hypernymy and other less-defined semantic relations (as shown in Table 2). Examples are:

– **Synonymy** (synonymy = identity):

5. *A companhia tem muitos funcionários...* ***Os trabalhadores*** *...*

The company has many employees... ***The workers*** *...*

– **Meronymy** (meronymy = part of or has parts):

6. *O computador apresenta inovações... principalmente,* ***o HD*** *...*

The computer presents innovations ... mainly, ***the HD*** *...*

– **Hypernymy** (hypernymy = generalization):

7. *A principal evidência era um memorando escrito pela polícia.* ***O documento*** *...*

The main evidence was a memorandum written by the police. ***The document*** *...*

– **Other semantic relations** (less-defined relations):

8. *O presidente da FIFA fala em aumentar o número de participantes na*

Copa de 32 para 37 países ... ***A idéia*** *...*
The president of FIFA speaks about increase the number of participants in the World Cup from 32 to 37 countries ... ***The idea*** *...*

Concerning the well-defined semantic relations of the *Associative anaphor*, meronymy prevailed. In *Indirect* cases, synonymy was dominant instead. This confirms formerly reported work [8, 7, 12].

4 Anaphor Resolution

Direct anaphors are usually the most common type, being basic surface constructions for text cohesion. Considering the nature of this type of anaphor, our first approach is to match the head-noun of the DD with the head-noun of the potential antecedent. The implemented Head-matching resolution algorithm is: Select all the DDs of the text; For each DD, identify its head-noun; Find a previous occurrence of the same head-noun; If one is found, return the corresponding noun phrase as the antecedent. As seen in Table 3, with this simple approach, 89% of the cases were correctly resolved. Other work consider various extensions of this approach, e.g., checking the distance of the (same-head) antecedent from the DD [6] and using the information provided by the premodifiers and postmodifiers of the noun phrase [11]. By including such extensions, it is likely that our results would improve. Errors are discussed next.

Problems related to unsolved cases were due to proper names (4), acronyms (4), and number variation (1). Proper names are grouped together when tokenized by the parser, with the use of underlines between their words. That prevents matching, e.g., between *Castor* and *Castor_de_Andrade*. An analogous situation occurs with acronyms, for example, *Produto_Interno_Bruto_-_PIB* does not match with *PIB*.

Non-matchable expressions in the plural and the singular form can refer to the same entity, as in 9. On the other hand, surface expressions with equal head nouns may refer to distinct entities (4 cases). Usually these have different modifiers (adjective or prepositional phrases), as in 10. Also problematic was the inclusion of new elements in the text progression (7 cases), as shown in 11.

9. *As crianças reclamam de maus tratos (...)* ***A criança*** *não tem acompanhamento familiar.*

Table 3. Head-matching Approach for Direct anaphora

Direct anaphora		
Results	**#**	**% of Total**
Correct	159	89%
Unsolved	9	5%
Error	11	6%
Total	**179**	**100%**

```
<word id="word_28">
<n canon="curso" gender="M" number="P">
  <secondary_n tag="sem-s"/>
  <secondary_n tag="per"/>
  <secondary_n tag="occ"/>
</n>
</word>
```

Fig. 1. Pos file given by PALAVRAS

The children complain of mistreatment (...) ***The child*** *has no family care.*

10. ***A placa da impressora*** *trabalha (...)* ***A placa do scanner*** *...*

The printer board *normally works (...)* ***The scanner board*** *...*

11. *Nos últimos cinco anos, o governo investiu 18% em educação. Segundo o UNICEF,* ***o índice*** *é baixo (...) No ano passado, foi registrado 75% de estudantes mulheres nas universidades, porém, esse ano,* ***o índice*** *aumentará, diz Travali.*

In the last five years, the government has invested 18% in education. According to UNICEF, ***the index*** *is low (...) In the last year, 75% of female students were registered at the university, but, this year,* ***the index*** *will increase, Travali said.*

Since *Indirect* and *Associative anaphors* involve semantic relationships, we exploited the semantic tags provided by the parser PALAVRAS. In 12, e.g., *cursos* [*courses*] is annotated with *sem-s*, *per* and *occ*, representing the features: "speak-work", "period of time" and "social event" (an occasion) respectively - Fig.1.

12. *O Eurocenter oferece* ***cursos*** *de japonês na bela cidade de Kanazawa.*

The Eurocenter offers Japanese ***courses*** *in the beautiful Kanazawa city.*

The semantic tags approach to AR consists in finding relationships with previous nouns through the semantic tags. The chosen antecedent will be the nearest expression with the largest number of equal semantic tags. For instance, in 13, two important relationships are resolved by applying this principle, the first, *japonês* - ***a língua*** (*Indirect anaphor*) and the second, *cursos de japonês* - ***As aulas do nível avançado*** (*Associative anaphor*). As both expressions *japonês* and *a língua* hold the semantic trace "idiom" the *Indirect anaphor* is resolved.

Table 4. Indirect and Associative anaphora for nouns

	Associative anaphora		**Indirect anaphora**		**Associative + Indirect anaphora**	
Results	**#**	**% of Total**	**#**	**% of Total**	**#**	**% of Total**
Correct	14	78%	29	54%	43	60%
Unsolved	1	5%	4	7%	5	7%
Error	3	17%	21	39%	24	33%
Total	**18**	**100%**	**54**	**100%**	**72**	**100%**

Table 5. Indirect and Associative anaphora for nouns and proper names

	Associative anaphora		**Indirect anaphora**		**Associative + Indirect anaphora**	
Results	**#**	**% of Total**	**#**	**% of Total**	**#**	**% of Total**
Correct	14	67%	29	47%	43	52%
Unsolved	4	19%	12	19%	16	19%
Error	3	14%	21	34%	24	29%
Total	**21**	**100%**	**62**	**100%**	**83**	**100%**

Table 6. Relations for nouns and proper names

	Synonymy		**Hypernymy**		**Meronymy**		**Other**	
Results	**#**	**% of Total**	**#**	**% of Total**	**#**	**% of Total**	**#**	**% of Total**
Correct	18	42%	12	80%	13	68%	0	0%
Unsolved	7	16%	0	0%	3	16%	6	100%
Error	18	42%	3	20%	3	16%	0	0%
Total	**43**	**100%**	**15**	**100%**	**19**	**100%**	**6**	**100%**

Table 7. Semantic tags approach for Direct anaphors

Direct anaphors		
Results	**# of DDs**	**% of Total**
Correct	93	52%
Unsolved	65	36%
Error	21	12%
Total	**179**	**100%**

Using the semantic tags, *sem-s*, *occ* and *per*, common to the expressions *cursos* and *aulas*, the *Associative anaphor* was also resolved.

13. *O Eurocenter oferece cursos de japonês na bela cidade de Kanazawa. (...)* ***As aulas do nível avançado*** *incluem refeições típicas... Após um mês, o aluno aplicado entenderá e falará modestamente* ***a língua***.

The Eurocenter offers Japanese courses in the beautiful Kanazawa city. (...) ***The advanced level classes*** *include typical meals ... After one month, a diligent student can modestly understand and speak* ***the language.***

We ran two experiments for *Indirect* and *Associative* AR, considering (a) nouns and proper names, and (b)just nouns. We consider as correctly resolved those cases in which the antecedent indicated in the manual annotation was in the anaphoric chain identified by the algorithm. As seen in Tables 4 and 5, *Associative anaphors* resolution has shown a better result than resolution of *Indirect anaphors*.

There was more unsolved cases (16) for proper names than unsolved ones in the first experiment (Table 4): actually 5 for nouns. This is due to the non-tagged proper names. Table 6 shows the error distribution over semantic relations. There are less errors for hypernymy and meronymy than for synonymy.

A third experiment, considering the commonest Direct Anaphors yielded the figures shown in Table 7. Compared with Table 3, semantic tagging for AR was worse than head-matching: unsolved cases amounted to 65 against 9 in the latter approach. This was also due to the lack of semantic tags for proper names and some nouns.

Most unsolved cases, in *Indirect* and *Associative* AR, are related to the fact that there is no semantic tagging for proper names in PALAVRAS as yet. So, e.g., the indirect anaphoric relationship between *São Carlos - a cidade* [*São Carlos - the city*] could not be resolved (a similar situation was observed for another 11 cases). Also, some nouns were not semantically annotated (5 cases), for instance *a propina* [*the bribe*].

Regarding wrong antecedents, some semantic relations between DDs and antecedents (2 cases) were not strong enough, yielding no semantic tags in common at all, as in *os rituais - o povo* [*the rituals - the people*] and *a proposta - o aumento* [*the proposal - the rise*]. Furthermore, some DDs (6 cases) referred to a whole sentence, paragraph or even disjoint parts of the text, as in 8. Since our approach considers only relations holding between noun phrases, these cases can not be resolved. Finally, there are cases (16) in which the nearest expression with the largest number of equal semantic tags is not the correct antecedent. For instance, establishing an *Indirect* relation between *os professores* [*the teachers*] - semantic tags *H*(human) and *Hprof* (professional human) - and *os politicos* [*the politicians*], with coinciding semantic tags *H* and *Hprof*, when the correct is *os professores* [*the teachers*] - semantic tags *H* and *Hprof* - and *os docentes* [*the docents*], with a different pair of semantic tags *HH* (group of humans) and *Hprof*. Overall, the low percentage of correct resolution for direct anaphora using semantic tags was mainly due to the lack of tags for proper names (53 cases), but also for acronyms (4) and some nouns (8).

5 Related Work

Previous work on AR has used lexical knowledge in different ways. Poesio et al. [8] aim at resolving bridging DDs ?, using the WordNet [2]. This work encloses our *Indirect* and *Associative anaphors* and reports a 35% recall rate for 12 cases of synonymy, 56% for 13 cases of hypernymy, and 38% for 11 cases of meronymy. Schulte im Walde [9] evaluated those cases through lexical acquisition on? the British National Corpus. She reported a recall of 33% for synonymy, 15% for hypernymy, and 18% for meronymy. Poesio et al. [7] also considered syntactic patterns for lexical knowledge acquisition, reporting the best results

Table 8. Related work Synonymy (Syn), Hypernymy (Hyp), Meronymy (Mer), Indirect anaphora (Ind), Associative anaphora (Ass), Coreference (Cor)

Works	Syn	Hyp	Mer	Ind	Ass	Cor
Poesio et al., 1997 (WordNet)	35%	56%	38%	-	-	-
Schulte im Walde, 1997	33%	15%	18%	-	-	-
Poesio et al., 2002	-	-	66%	-	-	-
Bunescu, 2003	-	-	-	-	23%	-
Gasperin and Vieira, 2004 (nouns)	-	-	-	33%	-	-
Gasperin and Vieira, 2004 (nouns and proper names)	-	-	-	22%	-	-
Markert and Nissim, 2005 (Web)	-	-	-	-	-	71%
Markert and Nissim, 2005 (WordNet)	-	-	-	-	-	65%
Our system (nouns and proper names)	42%	80%	68%	47%	67%	78%
Our system (nouns)	50%	80%	93%	54%	78%	-

for meronymy (66% recall). Gasperin and Vieira [3] tested the use of word similarity lists on resolving indirect anaphora. They performed two experiments, the first considering lists of nouns and the second, lists of proper names and nouns, reporting 33% of recall when just nouns were considered (57 cases), and 22% for proper names (95 cases). Markert and Nissim [4] presented two ways of obtaining lexical knowledge for antecedent selection in coreferent DDs (*Direct* and *Indirect anaphora*). They achieved 71% of recall using a Web-based method and 65% using a WordNet-based method instead.

Results of related work and ours are summarized in Table 8. We acknowledge that these experiments differ in data, languages and resources, therefore it is not a strict comparison. However, we consider that results were satisfactory on the light of related work, since by resolving 159 direct and 29 indirect anaphors, we were able to treat 78% of the coreferent cases. Note that usually evaluation on bridging anaphors is based on few cases, like most related work, because data are sparse. In our case only 21 out of 262 anaphoric DDs were associative.

There are several ways of looking at the above results, with respect to resolving anaphoric DDs to improve coherence in automatic summaries. Every other anaphoric relation presented in Section 2 may pose severe coherence and, thus, precision problems to TS, as pinpointed by [10] and others. The sort of focused problem is the following: if a span conveying a given DD were chosen to compose a summary and its antecedent span were not, there would be a referential break in the summary, which, in turn, would damage comprehension. This could happen, e.g., if we considered that the central idea of a summary derived from example 2 were related to the event *listening the police chief in charge*: the second sentence would be more likely to be included in the summary than the first, for its straighter relationship to the central idea. Potentially, the first text span could thus be excluded from the summary, yielding a non-resolved DD.

The commonest approaches to TS, namely, the shallow (knowledge-poor) and deep (knowledge-rich) ones, should tackle such problems. Considering the approach of importing to a summary full text spans (phrasal or sentential) of the source text, a shallow method would empirically or numerically manipulate information related to lexical items or phrases to find potential anaphoric chains. So, considering both approaches to anaphora resolution depicted in Section 4, guaranteeing the inclusion of a full anaphoric chain in a summary could be tackled by head-matching for direct anaphors and semantic tagging for indirect anaphors, if semantic information is provided.

6 Final Remarks and Future Works

We presented an evaluation of two different approaches to anaphoric DDs resolution, including direct, indirect and associative anaphors. The results show that semantic tagging by PALAVRAS is a good resource for dealing with the problem. They are encouraging, when compared to other work based on varied lexical resources, such as WordNet and acquired similarity lists. For direct anaphora, however, head matching proved better than that. The results shall improve once provided tagging for proper names. Our ongoing plan is to integrate both approaches and make a global evaluation. On this paper each DD class was reported separately and, then, the correctly resolved anaphors for each case were combined, resulting in 202 out of 262 anaphors correctly resolved. This gives us a score of 77% of correctly resolved anaphors for all anaphoric DDs. We also plan to improve AR through semantic tagging by considering other semantic aspects, such as a hierarchy of features, i.e., considering the influence of matching more specific features in opposition to more general ones in looking for antecedents.

Acknowledgments. This work was partially funded by CNPq, proc. nro. 50703020044.

References

1. E. Bick. *The Parsing System PALAVRAS: Automatic Grammatical Analysis of Protuguese in a Constraint Grammar Framework*. PhD thesis, Arhus University, Arhus, 2000.
2. C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.
3. C. Gasperin and R. Vieira. Using word similarity lists for resolving indirect anaphora. In *Proceedings of ACL Workshop on Reference Resolution and its Applications*, pages 40–46, Barcelona, 2004.

4. K. Markert and M. Nissim. Comparing knowledge sources for nominal anaphora resolution. *Computational Linguistics*, 31(3):367–401, 2005.
5. C. Müller and M. Strube. Mmax: A tool for the annotation of multi-modal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 45–50, Seattle, Washington, 2001.
6. M. Poesio, M. Alexandrov-Ksbadjov, R. Vieira, R. Goulart, and O. Uryupina. Does discourse-new detection help definite description resolution? In *Proceedings of the 6th International Workshop on Computational Semantics*, pages 236–246, Tiburg, 2005.
7. M. Poesio, I. Tomonori, S. Shulte im Walde, and R. Vieira. Acquiring lexical knowledge for anaphora resolution. In *Proceedings of 3rd Language resources and evaluation conference LREC 2002*, Las Palmas, 2002.
8. M. Poesio, R. Vieira, and S. Teufel. Resolving bridging descriptions in unrestricted texts. In *Proceedings of the Workshop on Operational Factors In Practical, Robust, Anaphora Resolution for Unrestricted Texts*, pages 1–6, Madrid, 1997.
9. S. Schulte im Walde. *Resolving Bridging Descriptions in High-Dimensional Space Resolving Bridging Descriptions in High-Dimensional Space*. PhD thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, and Center for Cognitive Science, University of Edinburgh, Edinburgh, 1997.
10. R. Stuckardt. Coreference-based summarization and question answering: A case for high precision anaphor resolution. In *Proceeding of the 2003 Intermnational. Symposium on Reference Resolution and Its Applications to Question Answering and Summarization*, pages 33–41, Venice, Italy, 2003.
11. R. Vieira. *Definite Description Processing in Unrestricted Text*. PhD thesis, University of Edinburgh, Edinburgh, 1998.
12. R. Vieira, S. Salmon-Alt, and C. Gasperin. Coreference and anaphoric relations of demonstrative noun phrases in multilingual corpus. *Anaphora Processing: linguistic, cognitive, and computational modelling*, 263:385–427, 2002.

Design of a Multimodal Input Interface for a Dialogue System

João P. Neto, Renato Cassaca, Márcio Viveiros, and Márcio Mourão

L^{2}F - Spoken Language Systems Laboratory,
INESC ID Lisboa / IST
Joao.Neto@l2f.inesc-id.pt
www.l2f.inesc-id.pt

Abstract. In this paper we described our initial work on the development of an embodied conversational agent platform. In the present stage our main focus it is on the development of a multimodal input interface to the system. In this paper we will present an Input and Output Manager block that combines speech, synthetic talking face, text and graphical interfaces. The system support speech input through an ASR and speech output through a TTS, synchronized with an animated face. The graphical and text input are feed through a Text Manger that it is a constituent component of the Input and Output Manager block. All the blocks are tailored for the European Portuguese language. The system is analyzed in the framework of the project Interactive *Home of the Future.*

1 Introduction

Human-computer conversation is a broad research goal that is starting to be implemented through a new genre of embodied conversational agents (ECA). All range of verbal and nonverbal behaviors seen in a face-to-face conversation are still far from reach. The models and behaviors necessary to natural conversation, in particular, emotion, personality, performatives, and conversational function, are starting to be implemented in these ECAs [1].

In this paper we described our initial work on the development of an ECA platform. In the present stage our main focus it is on the development of a multimodal input interface to the system.

In our laboratory we have been working on base technologies, as Automatic Speech Recognition (ASR), Text-to-Speech (TTS) and Natural Language Processing (NLP). These technologies have been applied in different environments, with diverse goals. Recently we developed a Synthetic Talking Face platform that it is closed integrated with these base technologies. Also we have been working on generic Spoken Dialogue Systems, integrating these base technologies. Due to the different application environments of the base technologies we end up with different application domains/tasks for Spoken Dialogue Systems.

The diversity of applications where we have been interested on is imposing some design restrictions to the overall architecture. We searched for a flexible architecture allowing different types of user interaction and access to the applications. This work end up in a system tailored to be task independent in the application specification [2] and a reconfigurable user interface.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 170 – 179, 2006.

In [2] there is a description of the main system features towards task independency and in [3] an analysis of the spoken interface robustness. In this paper we focus on technology analysis of the input interface block configuration and the contribution to the user interface efficiency.

To describe our system we will use mainly our participation in the project Interactive Home of the Future. This project is associated to an exhibition space that is intended to show a set of home environment gadgets. The visitor's feedback is indicative of the acceptability, the difficulties and the interest of a spoken interface. In order to analyze our input interface in the scope of a multimodal system we refer to the graphical interface that we have implemented in our demo room, which have some similar features as the exhibition space in a reduce dimension, confined to a single room [2].

In section 2 we will present the project of the Interactive Home of the Future including our participation. Section 3 gives a brief description of our spoken dialogue system, section 4 the multimodal input interface, section 5 the synthetic talking face and in 6 the user interaction with the system. The conclusions are presented in section 7.

2 The Project Interactive Home of the Future

The project Interactive Home of the Future has born in 2002 based on the initiative from FPC (Communications Portuguese Foundation [http://www.fpc.pt]).

This house is an advanced solution of Home-Automation integrating technologies ready available in different domains, from equipments, systems, furniture, design, infrastructure and building, and offer from telecommunications operators (cable TV, interactive TV, new telecommunications products), with some new prototypes being developed by Universities and research teams.

The idea was to show and disseminate to the generic public the present and future potentialities of the telecommunications and multimedia, create an offer of complementarities between different companies connected to the new technologies, and to contribute to the development and innovation in the area of communications and multimedia.

Professionals working in the area, under-graduated students, families and young students, are the most important groups that visit this exhibition.

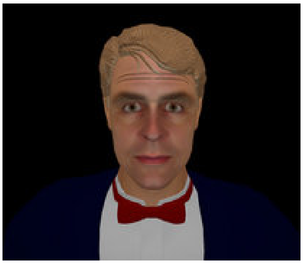


Fig. 1. – Synthetic talking face from the graphical interface of the system

The Spoken Language System Laboratory (L^2F) of INESC ID Lisboa was invited to participate in this project with the development of a spoken dialogue system. This system allows the home users to access through a spoken based interface to the different devices and services of the house. This system represents the concept of a virtual butler, someone that is always available to execute our requests. We combine automatic speech recognition, natural language understanding and speech synthesis. A visual interface based on a realistic animated synthetic face synchronized with the speech production process creates a good effect over the user helping in the dialogue process. The system responds to the name of "Ambrósio".

Our system operates on the different spaces of the house controlling most of the devices available. The devices typically are lights, window covers and TV sets, and some more specific as the opacity of a window glass from the bedroom to the hall. This situation allows us to show the real effectiveness of this system and its large applicability.

This project brings the possibility to work in a living laboratory to technology rehearsal, working over real problems and observe the users problems. Also was possible a strong engagement of students, contributing to their formation and the creation of a strong relationship with the companies involved in the project.

3 Our System

Our system is based on three main blocks (see Figure 2). Two of them are responsible for the interfaces with the user and the centralized system for control of devices (based on a web server). The other block is responsible for the dialogue management. The user interface is based on a wireless microphone and the TV sets, where "Ambrósio" is visualized and answer to our requests (the TV speakers are the output point of the speech generated by our system). The control of devices has an interface based on a web server, making available the access to any device in the home.

Despite this specific application of our system we have been developing these blocks in a more generic way, in order to cope with different types of applications. We use the same system to control the home environment, to access different databases (weather information, bus information, stock market information) and to email access. Also we get access to the system from microphone, telephone, GSM, PDA and web [2].

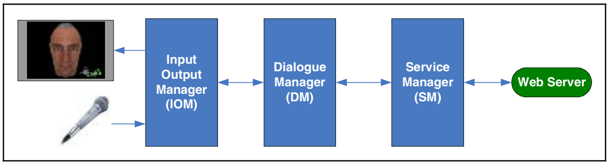


Fig. 2. - Block diagram of the overall system

In order to satisfy these goals both interface blocks create an independency level to the Dialogue Manager (DM). The Input and Output Manager (IOM) creates an independency level sending the same XML format, independent of the source. The same principle of independency is applied at the Service Manager (SM). The configuration of services creates a representation that is independent of the domain [2,4]. In this paper we are interested in the IOM block. In [3] there is a brief description of the other two blocks.

Input and Output Manager (IOM)

There are 4 main blocks in this diagram: the ASR, the TTS, the FACE and the TM.
The ASR is based on Audimus [5], a hybrid speech recognizer that combines the temporal modeling capabilities of Hidden Markov Models (HMMs) with the pattern discriminative classification capabilities of multilayer perceptrons (MLPs). This same recognizer is being used for different complexity tasks based on a common structure but with different components.

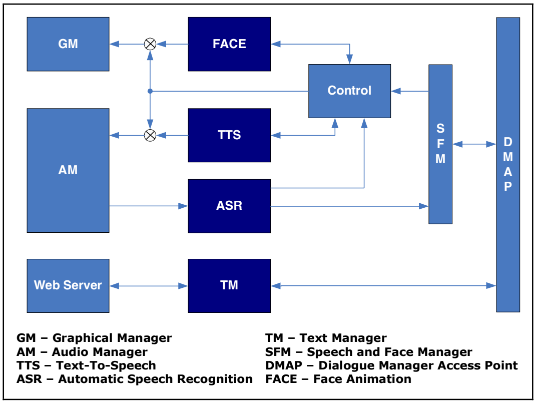


Fig. 3. - Block diagram of the Input and Output Manager

The TTS module (DIXI+) [6] is a concatenative-based synthesizer, based on the Festival framework. This framework supports several voices and two different types of unit - fixed length units (such as diphones), and variable length units. This latter data-driven approach was suitable, by adequate design of the corpus, to a limited domain of application as our present situation.

The FACE [7] is a Java 3D implementation of a synthetic talking face with a set of visemes for the Portuguese phonemes and a set of emotions and head movements. In section 5 we present a more detailed description of this module.

The TM transforms the web server requests, from PDA and web, in the XML format to access the DM.

The CONTROL block receives an XML file, with text, emotions and head movement's descriptions, splitting and feeding the appropriate information to the FACE and TTS. This block is responsible to the synchronization of these systems outputs.

The SFM manages the interface between the speech and animated face blocks with the DSAP. It is a simple block that encapsulates the ASR output in a XML format and, in the other direction, responsible to send the XML file, received from the DM, to the CONTROL block.

DSAP manages the communication between the SFM and TM with the DM. DM uses the hub structure of the Galaxy II.

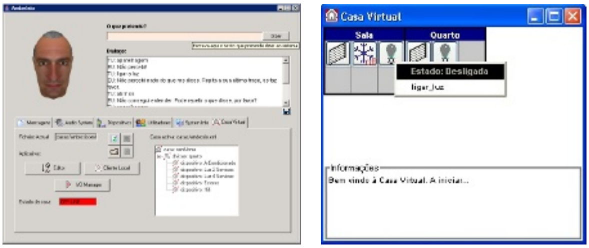
4 Multimodal Input Interface

The IOM block, represented in Fig. 3, is the key component to enlarge the user system accessibility. The communication with the user is based on a Graphical Manager (GM), an Audio Manager (AM) and a Web Server (WS). The GM is only an output device where the animated agent face is represented. The AM is designed to deal with several audio devices, as mic/speakers (including bluetooth and wireless devices) and fixed/mobile telephone. Based on this AM the system is configurable to be locally accessible, in a room, or remotely by phone. The Web Server allows local communication by a PC or PDA, or again remotely by any PC in the web.

As a text interface we designed a block that is connected to the hub structure that interconnects the IOM, the DM and SM, from Fig. 2. This block is used mainly for debug purposes and system configuration. In Figure 4 we show the design of both interfaces.

The spoken input is feed in the IOM through the AM, transcribed by the ASR, transformed in a frame by the SFM block and sent to the DM through DMAP. The graphical input to the system is transformed in a text string corresponding to the graphical action at the client application, and transmitted to the Web Server. At the TM that string it is transformed into a frame and sent to the DM through the DMAP, as in the spoken input. The text input could also be connected to the TM but we decided to make it directly available in the hub, mainly due to debug purposes since we are able to see all the messages in the hub and the possibility to easily configure all the three main blocks.

The fusion process is taking place at the DM on a first-come-first-serves basis.



Bem vindo à Casa Vitual. A iniciar...

a) b)

Fig. 4. - Interfaces for: a) text input/output, debug and configuration; b) graphical input

5 Synthetic Talking Face

Human Computer Interface is an area where audio, text, graphics, and video are integrated to convey various types of information. The goal is to provide a more natural interaction between the user and the computer.

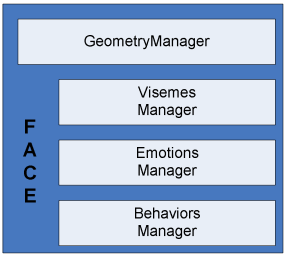


Fig. 5. - Facial Animation System structure

One approach to achieve this goal is to display an animated character on the computer screen, with the ability to make head movements, facial expressions and emotions. When associated with speech, the overall facial expressions constitute one of the most important communication channels used in human interactions.

Facial expressions allow the exposure of these emotions that play an important role in the context of human communication. The ability of expressing feelings like sadness, happiness or hanger allows the machine to emulate human emotions, acquiring capabilities only seen in humans, and bringing more reality to Human-Machine interactions.

In our system the FACE block on the IOM (see Fig. 3) produces facial expressions through the implementation of virtual muscles. When this muscles are stimulated, they deform a three dimensional mesh resulting in expressions.

Mathematic models simulating muscles behaviors were defined and associated with regions containing vertices of a polygonal mesh representing the human face. Parameters such as the intensity of contraction and time of reaction are passed to these virtual muscles. Using these parameters, the muscles act in the vertices region, resulting in deformations on the surface model. To simplify the muscles manipulation, groups of muscles were defined. These groups are associated to visemes and emotions. This way when one or more groups are activated, we get a facial expression that represents a viseme, an emotion or both. A viseme is the visual representation of a phoneme and is usually associated with muscles positioned near the region of the mouth. Emotions, in this project, are expressions simulating the real human emotions such as fear, joy or sadness and can be associated with any muscle on the face.

To create the animations in real time we feed the system with sequences of phonemes, which are then transformed in visemes, and sequences of emotions and behaviors. These sequences are then combined and transformed in key-frames. A key-frame is used as a reference, in this case is used for calculate the intermediate frames using interpolation methods. These animations are described in the form of a simplified VHML (Virtual Human Markup Language) [8], whose structure was defined according to the project goals.

The facial animation system is composed by 4 modules: (i) the Visemes Manager; (ii) the Emotions Manager; (iii) the Behavior Manager; and (iv) the Geometry Manager (see Fig. 5).

5.1 Visemes Manager

Visemes are visual representation of phonemes. To be able to represent visemes on a 3D face model, some video analysis has been done, allowing for the capture of values on the mouth movement during the phonemes pronunciation. These values represent the 3 distances A, B and C depicted in Fig. 6.

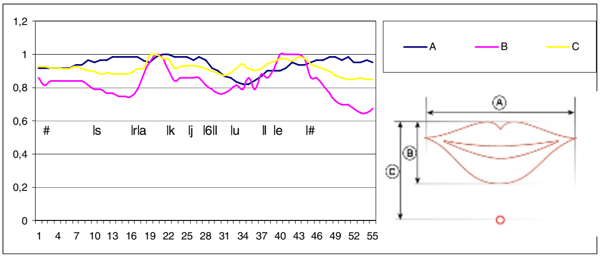
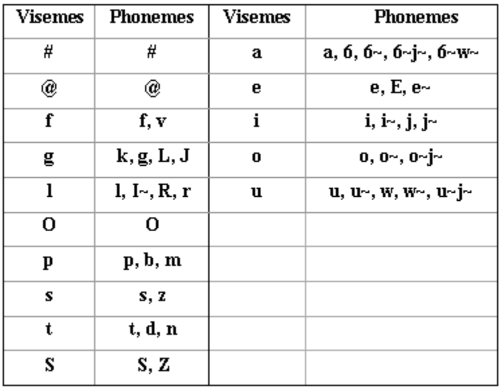


Fig. 6. – Distances extracted from video analysis. The graph (on the left) represents the distance after normalization per frame.

Table 1. – Relation between Visemes and Phonemes for the European Portuguese



Visemes	Phonemes	Visemes	Phonemes
#	#	a	a, 6, 6~, 6~j~, 6~w~
@	@	e	e, E, e~
f	f, v	i	i, i~, j, j~
g	k, g, L, J	o	o, o~, o~j~
l	l, l~, R, r	u	u, u~, w, w~, u~j~
O	O		
p	p, b, m		
s	s, z		
t	t, d, n		
S	S, Z		

The visemes manager module receives a sequence of phonemes and their ending times and transforms phonemes in visemes using Table 1 relationships. The module then builds key-frames that contain: (i) the muscles associated with each viseme; (ii) the intensity to apply to the muscles; and (iii) timing information. The intensity values are calculated based on the distances captured previously.

5.2 Emotions Manager

Given the sequence of emotions, this module creates and returns a list of key-frames containing the muscles associated with the emotions, the intensity to apply to those muscle and timing info.

5.3 Behaviors Manager

The goal of this module is to improve the quality of the animations, making them more realistic. Examples of animations are: head look, eyes movement and eye blink. Fig. 7 shows some results on movements applied to the face model.

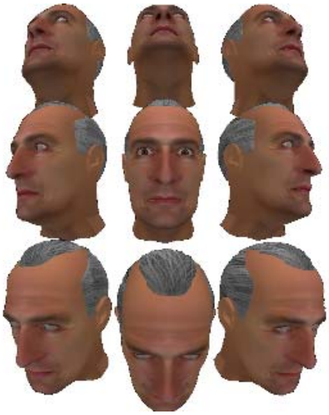


Fig. 7. – Some possible movements produced by this module

5.4 Geometry Manager

This is the core module of the system. The Geometry Manager is responsible for all the geometric manipulation applied to the three-dimensional face model.

The muscles developed for this project are based on the models proposed by Waters [9]: (i) linear muscle; and (ii) sphincter muscle. As explained before these muscles are created, and associated with the geometric model using references to regions that contain the vertices that describe the surface model. These regions can be configured at any time. Once created and associated to the region, the muscles are stimulated through parameters such as the contraction intensity and time of response.

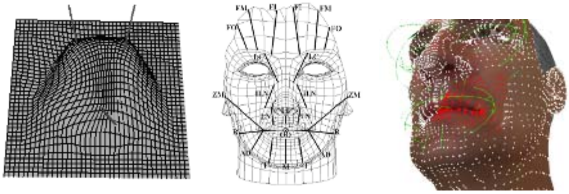


Fig. 8. – (a) Example of muscles deformation. (b) Muscles and facial mesh association. (c) Polygonal meshes and muscles region.

6 User interaction with the System

Microphone

In the design process of the spoken interface there was a long discussion about the type of microphone. A head-mounted microphone is preferable in terms of speech recognition performance but it is intrusive. A wireless shirt mounted microphone is a good alternative but introduces a significant degradation in the signal quality due to omni directionality. Another approach is the use of a set of microphone arrays mounted on the room ceiling or wall. There are localization, reverberation and cross-talk problems with this approach, besides the high cost. In our home environment and after some informal tests we verify that the users prefer to talk to a specific point or device and not to the "air". At the end we have chosen as appropriate a wireless hand microphone. Despite you have to carry it was not so intrusive. It was also clear who have the power to control the devices. We also take the option of keeping always the microphone open and it is the system that through keyword identification starts the recognition process.

ASR configuration

At the input of the ASR we activate a keyword detection block implementing a kind of efficient DTW algorithm to filter the input. This means that to control the devices the user has to start by the keyword "Ambrósio". The acoustic models of our Audimus [5] system are speaker independent.

The vocabulary and language model, due to our goal of task independency, should be automatically extracted from the SM configuration XML files. Presently this feature is not full implemented. We create a vocabulary from the devices functions and from an XML description of the possible interactions with that device. To enrich the dialogue we add other words that we consider generic and task independent. We are using BNF grammars and n-gram language models.

Graphical and text interface

Besides the spoken interface we have the two other input modalities. In the text interface we write through the keyboard the command to the system and we get the answer. This interface is mainly use for debug process. The graphical interface presents to the user the configuration of the devices divided by rooms. The user has to click on each device to change the state (see Fig. 4b).

7 Conclusions

The use of the system has been showing that the visitors prefer the use of the spoken interface due to their naturalness and facilities, and are willing to repeat the command in system failure situations, due to ASR errors. The graphical interface is very simple but the users only use it if the spoken interface repeatedly fails to respond correctly. A multimodal interface including speech and graphics needs to be correctly designed in order to take full advantage of the input possibilities. As a result of our participation in the Interactive Home of the Future project it was possible to show to the general public, with great success, the advantages of a spoken interface.

Acknowledgements

This work was partially funded by FCT project POSI/PLP/41319/2001 (POSI and FEDER). There are other colleagues that contributed to this project, as A. Quintino, N. Pedro, Sérgio Paulo, Nuno Mamede and Luís Oliveira.

References

[1] J. Cassell, J. Sullivan, S. Prevost and E. Churchil (eds.), “Embodied conversational agents”, MIT Press, 2000.

[2] J. Neto, N. Mamede, R. Cassaca, L. Oliveira, "The development of a multi-purpose Spoken Dialogue System", Proc. Eurospeech 03, Genéve, Swiss, 2003.

[3] J. Neto, R. Cassaca, "A Robust Input Interface in the scope of the Project Interactive Home of the Future", Proc. ROBUST 04, Norwich, UK, 2004.

[4] M. Mourão, R. Cassaca and N. Mamede, "An independent domain Dialogue System through a Service Manager", Proc. ESTAL 2004, Alicante, Spain, 2004.

[5] H. Meinedo, D. Caseiro, J. Neto and I. Trancoso, "AUDIMUS.MEDIA a Broadcast News speech recognition system for the European Portuguese language", Proc. PROPOR'03, Faro, Portugal, 2003.

[6] S. Paulo and L. Oliveira, "Multilevel Annotation Of Speech Signals Using Weighted Finite State Transducers", Proc. 2002 IEEE Workshop on Speech Synthesis, Santa Monica, USA, 2002.

[7] M. Viveiros, "Cara Falante", Graduation Thesis, IST.

[8] http://www.vhml.org

[9] K. Waters. A Muscle Model for Animating Three-Dimensional Facial Expressions. Computer Graphics (ACM SIGGRAPH'87), 21(4):17-24, July 1987.

Review and Evaluation of DiZer – An Automatic Discourse Analyzer for Brazilian Portuguese

Thiago Alexandre Salgueiro Pardo and Maria das Graças Volpe Nunes

Núcleo Interinstitucional de Lingüística Computacional (NILC),
CP 668 – ICMC-USP, 13.560-970 São Carlos, SP, Brasil
taspardo@gmail.com, gracan@icmc.usp.br
http://www.nilc.icmc.usp.br

Abstract. This paper presents the review and evaluation of DiZer – an automatic discourse analyzer for Brazilian Portuguese. Based on Rhetorical Structure Theory, DiZer is a symbolic analyzer that makes use of linguistic patterns learned from a corpus of scientific texts to identify and build the discourse structure of texts. DiZer evaluation shows satisfactory results for scientific texts. In order to test its portability, DiZer is also evaluated with news texts and presents acceptable performance.

1 Introduction

Researches in Linguistics and Computational Linguistics have shown that a text is more than just a simple sequence of juxtaposed sentences. It has a highly elaborated underlying discourse structure. In general, this structure represents how the information conveyed by the text propositional units (i.e., the meaning of the text segments) correlate and make sense together.

The ability to derive discourse structures of texts automatically is of great importance to many applications in Computational Linguistics. For instance, it may be very useful for automatic text summarization (to identify the most important information of a text to produce its summary) (see, for instance, O'Donnel, 1997; Marcu, 2000), co-reference resolution (determining the context of reference in the discourse may help determining the referred term) (see, for instance, Cristea et al., 1998; Schauer, 2000), and for other natural language applications as well.

Some discourse analyzers are already available for both English (e.g., Marcu, 1997, 2000; Corston-Oliver, 1998; Soricut and Marcu, 2003) and Japanese languages, (e.g., Sumita et al., 1992). For English, Marcu's analyzer was the first one available and was developed for free domain texts (based on news texts). To our knowledge, for Brazilian Portuguese, DiZer (DIscourse analyZER) (Pardo et al., 2004) is the only automatic analyzer for this language. Based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1987), DiZer is a symbolic analyzer that makes use of linguistic patterns learned from a corpus of scientific texts from Computer Science domain to identify and build the discourse structure of texts. Basically, DiZer follows the analysis strategy proposed by Marcu (1997, 2000), using cue-phrases occurrences in a text to build its discourse structure.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 180 – 189, 2006.

In this paper, we review DiZer main aspects and present a comprehensive evaluation of the system. We describe the construction of a reference rhetorically annotated corpus, called Rhetalho (Pardo and Seno, 2005), and the annotation protocol followed by human judges in order to achieve agreement. DiZer evaluation based on Rhetalho is presented and discussed for both scientific and news texts. Results show that DiZer performance is satisfactory.

Firstly, in the next section, we introduce RST, the discourse theory that DiZer follows. In Section 3, DiZer main processes and information repositories are reviewed. Section 4 describes DiZer evaluation procedure and results. Some conclusions and final remarks are made in Section 5.

2 Rhetorical Structure Theory

There are several discourse theories that try to represent different aspects of discourse (see, e.g., Grosz and Sidner, 1986; Mann and Thompson, 1987; Jordan, 1992; Kehler, 2002). Rhetorical Structure Theory (RST) (Mann and Thompson, 1987) is one of the most used theories and underlies most of the existent automatic discourse analyzers.

According to RST, all propositional units in a text must be connected by rhetorical relations in some way for the text to be coherent. The connection of all the text propositional units produces its rhetorical/discourse structure. Rhetorical structures are usually represented by trees (not necessarily binary), with each relation connecting subtrees, which can be single propositional units (that are leaves in the tree) or other trees. As an example of a rhetorical analysis of a text, consider Text 1 in Figure 1 (with segments that express basic propositional units numbered) and its rhetorical structure in Figure 2.

[1] Although he is allergic to it, [2] he tried it. [3] Now, he has a headache and [4] his body is red.

Fig. 1. Text 1

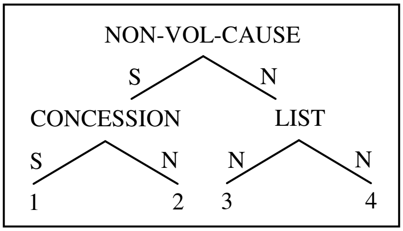


Fig. 2. Text 1 rhetorical structure

The symbols N and S indicate the nucleus and satellite of each rhetorical relation: in RST, the nucleus indicates the most important information in the relation, while the satellite provides complementary information to the nucleus. In this structure, propositions 1 and 2 are in a CONCESSION relation, i.e., the fact of being allergic to something should avoid someone of trying it; propositions 1 and 2 CAUSE (not volitionally) propositions 3 and 4; propositions 3 and 4 present a LIST of allergy symptoms. In some cases, relations are multinuclear (e.g., LIST relation), that is, they have no satellites and the connected propositions have the same importance; otherwise, relations are mononuclear, with one nucleus and one satellite (e.g., CONCESSION and NON-VOL-CAUSE relations). RST originally defines around 25 relations.

One last point about RST that must be mentioned is that, in order to guarantee the construction of valid and well-formed rhetorical structures during the analysis of texts, Mann and Thompson established the compositionality criterion. It says that, for connecting two subtrees T1 and T2 by a relation R in order to form a bigger tree T3, R must hold between the most salient propositional units of T1 and T2, i.e., R must relate the most nuclear units of subtrees T1 and T2. For example, in Figure 2, to form the complete tree, the NON-VOL-CAUSE relation must hold between the most salient units of the subtrees headed by the CONCESSION and LIST relations, i.e., it must connect units 2 (from the left subtree) and 3 and 4 (from the right subtree). If in the text NON-VOL-CAUSE would relate units 1 (which is a satellite from the left subtree and, therefore, is not the most salient unit in the subtree) and units 3 and 4 (from the right subtree), the structure in Figure 2 would be an invalid structure, since it would violate the criterion. This will be further discussed in Section 4.

As in other automatic discourse analysis works, RST is the discourse theory followed by DiZer, which is reviewed in the next section.

3 DiZer

DiZer comprises three main processes: (1) the segmentation of the text into propositional units, (2) the detection of occurrences of rhetorical relations between propositional units and (3) the building of the rhetorical structures. Figure 3 presents the system architecture. In the next subsections, each process is explained. The information repositories are introduced as the processes that use them are explained.

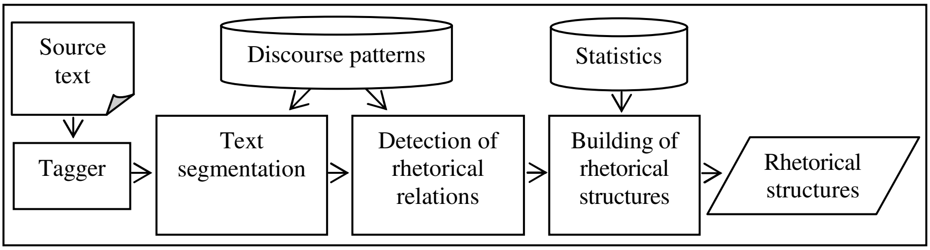


Fig. 3. DiZer architecture

3.1 Text Segmentation

In this process, DiZer tries to determine the simple clauses in the source text, since simple clauses usually express single propositional units, which are assumed to be the minimal units in a rhetorical structure. For doing this, DiZer initially attributes morphosyntactic categories to each word in the text using a Brazilian Portuguese tagger (Aires et al., 2000). Then, the segmentation process is carried out, segmenting the text always a punctuation signal (comma, dot, exclamation and interrogation points, etc.) or a strong cue phrase is found. Given the ambiguity of dot, an abbreviation list is used to identify which dots are sentence boundaries. By strong cue phrase, we mean those words that unambiguously have a function in discourse, clearly indicating a rhetorical relation between propositions or signaling the discourse structure. According to this, words like *e* and *se* (in English, “and” and “if”, respectively) are ignored, while words like *portanto* and *por exemplo* (in English, “therefore” and “for instance”, respectively) are not. The cue phrases are retrieved from the “Discourse patterns” repository, which is better explained in the next subsection. DiZer still verifies whether the identified segments are clauses by looking for occurrences of verbs in them. Optionally, in DiZer, it is also possible to perform sentence segmentation instead of clause segmentation.

3.2 Detection of Rhetorical Relations

DiZer tries to determine at least one rhetorical relation for each two adjacent text segments representing the corresponding underlying propositions. Initially, it looks for a relation between every two adjacent clauses inside a sentence; then, it considers every two adjacent sentences of a paragraph; finally, it considers every two adjacent paragraphs. This processing order is supported by the premise that a writer organizes related information at the same organization level. For instance, if two propositions are directly related (e.g., a cause and its consequence), it is probable that they will be expressed in the same sentence or in adjacent sentences.

Relation	OTHERWISE
Order	satellite (S) before nucleus (N)
Marker1	---
Position of marker1	---
Marker2	*ou, alternativamente,*
Position of marker2	beginning

Fig. 4. Discourse pattern for the OTHERWISE rhetorical relation

Relation	PURPOSE
Order	satellite (S) before nucleus (N)
Marker1	---
Position of marker1	---
Marker2	*cujo_lem purWord _adj ser_lem*
Position of marker2	beginning

Fig. 5. Discourse analysis pattern for the PURPOSE rhetorical relation

In order to look for rhetorical relations, DiZer makes use of linguistic patterns stored in the “Discourse patterns” repository. Each pattern codifies the possible rhetorical relations that cue phrases may indicate. As an example, consider the discourse pattern for the OTHERWISE rhetorical relation in Figure 4. According to it, an OTHERWISE relation connects two segments 1 and 2, with 1 being the satellite and 2 the nucleus and with the segment that expresses 1 appearing before the segment that expresses 2 in the text, if the cue phrase *ou, alternativamente,* (in English, “or, alternatively,”) be present in the beginning of the segment that expresses propositional unit 2.

The discourse patterns may also convey morphosyntactic information, lemma and specific genre-related information. For instance, consider the pattern in Figure 5, which hypothesizes a PURPOSE relation. This pattern specifies that a PURPOSE rhetorical relation is found if there is in the text an cue phrase composed by (1) a word whose lemma is *cujo* (“which”, in English), (2) followed by any word that indicates purpose (represented by the “purWord” class), (3) followed by any adjective, (4) followed by a word whose lemma is *ser* (verb “to be”, in English).

For detecting the relations, DiZer performs a pattern matching process between text segments and the discourse patters.

For relations that are not explicitly signaled by cue phrases, like EVALUATION and SOLUTIONHOOD, DiZer uses heuristics. For the SOLUTIONHOOD relation, for example, the following heuristic holds:

> if in a segment X, 'negative' words like 'cost' and 'problem' appear more than once and, in segment Y, which follows X, 'positive' words like 'solution' and 'development' appear more than once too, then a SOLUTIONHOOD relation holds between propositions expressed by segments X and Y, with X being the satellite and Y the nucleus of the relation

When more than one rhetorical relation is detected for two segments, usually in occurrences of ambiguous or multiple cue phrases, all the possible relations are considered. Because of this, several discourse structures may be produced for the same text. In the worst case, when no rhetorical relation can be found between two segments, DiZer assumes a default heuristic: it adopts an ELABORATION relation, with the segment that appears first in the text being its nucleus.

The discourse patterns and heuristics were produced by manually annotating and analyzing a corpus of 100 Computer Science scientific texts in Brazilian Portuguese, called CorpusTCC (Pardo and Nunes, 2003, 2004). More details about this corpus and the knowledge extraction process to produce the patterns and heuristics can be found in Pardo et al. (2004).

3.3 Building of Rhetorical Structures

This process consists in determining the complete rhetorical structure from the individual rhetorical relations between the text segments. For this, the system makes use of the rule-based algorithm proposed by Marcu (1997). This algorithm produces grammar rules for each possible combination of segments by a rhetorical relation, in the form of a DCG (Definite-Clause Grammar) rule (Pereira and Warren, 1980). When the grammar is executed, all possible valid rhetorical structures are built.

Marcu’s algorithm incorporates the compositionality criterion established by RST (see Section 2). In DiZer, this criterion is ignored when it shows to be to restrictive to allow the production of any rhetorical structure, as will be discussed in the next section.

In the end of this process, DiZer offers the possibility of ranking all the produced structures by their probabilities. The probability of a structure is simply the multiplication of the probabilities of each relation and their immediate children (with their nuclearity indication) in the tree, which can be other relations or leaves (if they are terminal nodes). These probabilities are simple frequency counts collected from CorpusTCC and are stored in the “Statistics” repository in the form of conditional probabilities (i.e., the probability of the children and their nuclearity given the parent). When a probability is required and is not found in the repository, a very low probability (which was empirically established as 10^{-6}) is used, guaranteeing that the rhetorical structure have a non-zero probability.

As a complete example of DiZer processing, Figures 6 and 7 present, respectively, a text (in Portuguese) already segmented by DiZer and one of the valid rhetorical structures built. One may verify that the structure is totally plausible.

> [*Desde a sua abertura comercial, em 1993, a Internet tornou-se um meio de comunicação poderoso,*]$_1$ [*ao permitir a um usuário entrar em contato com quaisquer outros, espalhados pelo mundo todo.*]$_2$
>
> [*O comércio eletrônico é um dos novos nichos de exploração comercial da rede mundial de computadores,*]$_3$ [*pois ela torna possível realizar transações comerciais de forma global, com custo de manutenção inferior ao empregado em uma rede de comércio tradicional.*]$_4$
>
> [*O objetivo deste trabalho é apresentar uma proposta para o projeto e implementação de um serviço de comércio eletrônico na plataforma JAMP.*]$_5$ [*Esta plataforma constitui-se em um middleware implementado em Java/RMI para desenvolvimento de aplicações multimídia distribuídas, e em particular, aplicações para World Wide Web (WWW), através de frameworks de serviços para suporte ao desenvolvimento destas aplicações.*]$_6$

Fig. 6. Text 2

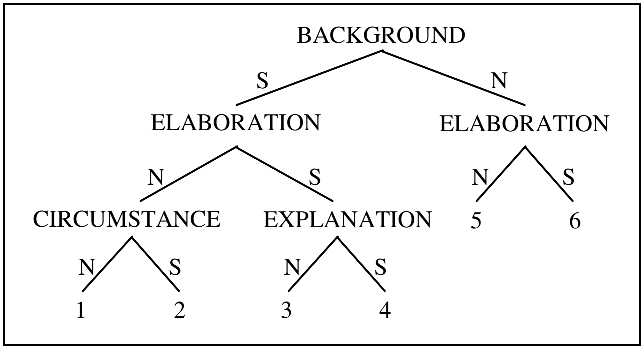


Fig. 7. Text 2 rhetorical relation

The probability of the structure in Figure 7 is given by:

$$\begin{aligned}
\text{P(structure)} = {} & \text{P(ELABORATION,S,ELABORATION,N|BACKGROUND)} \times \\
& \text{P(CIRCUMSTANCE,N,EXPLANATION,S|ELABORATION)} \times \\
& \text{P(leaf,N,leaf,S|ELABORATION)} \times \\
& \text{P(leaf,N,leaf,S|CIRCUMSTANCE)} \times \\
& \text{P(leaf,N,leaf,S|EXPLANATION)}
\end{aligned}$$

Next section describes DiZer evaluation.

4 Evaluation

In order to objectively evaluate DiZer, a reference corpus was produced. The corpus, called Rhetalho (Pardo and Seno, 2005), is composed of 50 rhetorically annotated texts (with size around half a page) from scientific and news genres, which are not in CorpusTCC. The scientific texts are from Computer Science domain; the news texts were collected from diverse sections from the on-line newspaper *Folha de São Paulo*.

All the texts were annotated by two judges (experts in RST), using Daniel Marcu's RST Annotation Tool (available at http://www.isi.edu/~marcu/discourse/) and following an annotation protocol in order to achieve agreement. The protocol specifies the following:

- the annotation of a text must be linear, from left to right, modular and incremental; by modular, it means that clauses inside sentences must be related first, then sentences inside paragraphs must be related, and, finally, paragraphs must be related; by incremental, it means that, whenever possible, as soon as a new segment is determined, it must be related to the subtree already built until that point;
- only binary structures are allowed, i.e., each node in the tree may have 2 children at most; with this, when a non-binary tree is produced, it must be transformed in a binary tree (for instance, a CONTRAST relation with 3 children should be transformed in a CONTRAST relation with 2 children, with one being the first child and the other being another CONTRAST relation connecting the 2 remaining children);
- for segmenting a text, the rules defined by Carlson and Marcu (2001) must be followed (although they were defined for the English language, they are generic enough to be applied to Portuguese too); when the judges disagree about a segment, the most comprehensive segment must be chosen;
- when judges hypothesize different relations for connecting two segments, the most generic one must be chosen; when they are equally generic and plausible, a third judge must be consulted.

DiZer was evaluated with 20 scientific texts and 5 news texts (from Section World) randomly selected from Rhetalho. The evaluation with news texts was conducted in order to verify the possibility of using DiZer with other text genres and domains, since it was developed based only on a corpus of Computer Science scientific texts.

Recall and precision were computed for the main aspects of the rhetorical structures produced by DiZer, namely, delimited segments, nuclearity of segments and detected rhetorical relations. This was done for both clausal and sentential segmentation in DiZer. For text segmentation, recall indicates how many segments of the reference structure (from Rhetalho) were correctly delimited and precision indicates how many of the delimited segments were correct; for nuclearity of segments, recall indicates how many nucleus and satellites of the reference structure were correctly identified and precision indicates how many of the segments were correctly classified (as nuclei or satellites); for rhetorical relations detection, recall indicates how many relations between segments of the reference structure were correctly detected and precision indicates how many of the detected relations were correct.

Table 1. DiZer performance for scientific texts

	DiZer – sentential segmentation (%)			DiZer – clausal segmentation (%)			Baseline method (%)		
	R	P	F	R	P	F	R	P	F
Segmentation	25.2	41.7	31.4	57.3	56.2	56.8	25.2	41.7	31.4
Nuclearity	39.1	69.5	50.1	79.7	82.3	80.9	32.4	59.5	42.0
Relations	28.7	61.0	39.1	63.2	61.9	62.5	20.7	49.2	29.2

Table 2. DiZer performance for news texts

	DiZer – sentential segmentation (%)			DiZer – clausal segmentation (%)			Baseline method (%)		
	R	P	F	R	P	F	R	P	F
Segmentation	9.9	20.6	13.4	48.8	54.1	51.3	9.9	20.6	13.4
Nuclearity	22.3	55.3	31.8	55.8	63.5	59.4	28.4	71.3	40.7
Relations	12.5	38.3	18.9	37.8	43.2	40.3	17.6	58.3	27.0

In order to judge DiZer results validity, we run the same evaluation for a baseline method. The baseline method performs sentential segmentation and detects only ELABORATION relations (given that the ELABORATION relation is usually the most generic and frequent one in texts), with the first segment being the nucleus.

Table 1 presents the resulting recall (R) and precision (P) average numbers for the baseline method and for DiZer analyses with clausal and sentential segmentation for scientific texts. Table 2 presents the numbers for the news texts. F-measure (F), which is a combination of recall and precision, is also showed. It is a unique measure of how good a system is.

According to the f-measures, for scientific texts, DiZer outperformed the baseline method for both sentential and clausal segmentation, with very good results for the latter. For the news texts, DiZer outperformed the baseline method for the clausal segmentation only. We believe that DiZer bad results for sentential segmentation with news texts are due to the way news texts are organized: most of the relations in news texts are ELABORATION, with the first segment being the nucleus, which is exactly the way the baseline method works.

In general, the clausal segmentation outperforms the sentential segmentation because it enables DiZer to produce more fine-grained structures, which are closer to Rhetalho reference structures.

DiZer performance shows to be satisfactory (even for news texts, when clausal segmentation is carried out, overcoming the baseline method). It also conforms to other works results, in particular, to Marcu's analyzer (1997, 2000), which is the most similar to DiZer in literature. Although this direct comparison is unfair, given that the languages and test corpora differ, it gives an idea of the state of the art results in cue-phrase-based analyzers.

In relation to the errors committed by DiZer, we identified some of the reasons that caused them. In clausal segmentation, the lack of a syntactic parser does not allow the exact determination of clause boundaries; simple rules based on punctuation signals are not enough for achieving very good results. In rhetorical relations detection, most of segments do not contain cue phrases, which causes the generation of a big amount of ELABORATION relations. Still, if the tagger fails in identifying the morphosyntactic classes of words, discourse analysis may be compromised during clausal segmentation (if verbs are not correctly classified) and rhetorical relations detection (when a discourse pattern asks for morphosyntactic classes that may be wrong in the sentence). Another problem, not so frequent in our test corpus, is related to the quality of the text to be analyzed: in some cases, cue phrases are misused, which introduces errors during rhetorical relations detection.

During DiZer evaluation, we also verified how many times the compositionality criterion could be applied. For scientific texts, the criterion was applied in 75% of the cases for sentential segmentation and in only 20% of the cases for clausal segmentation; for news texts, the criterion was applied in 60% of the cases for sentential segmentation and in only 20% of the cases for clausal segmentation. If DiZer were unable to ignore the compositionality criteria when this was too restrictive to allow the production of any rhetorical structure, just a few texts would have their structures produced. In general, we found that the compositionality criterion is desired in theory, but, in an automatic analyzer, it may not be: a single relation or nuclearity that is wrongly hypothesized for a text (which happens frequently in automatic discourse analysis, given the subjectivity of texts) may avoid the construction of any structure. In addition to this, Pardo (2005) shows that it is possible to have plausible rhetorical structures even when the compositionality criterion is not applied.

Next section presents some conclusions and makes some final remarks.

5 Conclusion

This paper reviewed DiZer main aspects and presented a comprehensive evaluation of the system, which showed satisfactory results. To our knowledge, DiZer is the first discourse analyzer for Brazilian Portuguese.

Although DiZer was developed for scientific texts analysis, its evaluation shows that it is possible to achieve acceptable results for other text genres and domains. We believe that this happens because, in general, cue phrases are consistently used by people in any kind of text.

DiZer is the first step towards the automation of other levels of discourse analysis. As suggested by Pardo (2005), it is possible to map directly rhetorical relations to the semantic relations proposed by Kehler (2002). This should be investigated in the future.

Acknowledgments

The authors are grateful to FAPESP, CAPES, CNPq, and to Fulbright Commission for supporting this work.

References

Aires, R.V.X.; Aluísio, S.M.; Kuhn, D.C.S.; Andreeta, M.L.B.; Oliveira Jr., O.N. (2000). Combining Multiple Classifiers to Improve Part of Speech Tagging: A Case Study for Brazilian Portuguese. In the *Proceedings of the Brazilian AI Symposium* – SBIA, pp. 20-22.

Carlson, L. and Marcu, D. (2001). *Discourse Tagging Reference Manual.* ISI Technical Report ISI-TR-545.

Corston-Oliver, S. (1998). *Computing Representations of the Structure of Written Discourse*. PhD Thesis, University of California, Santa Barbara, CA, USA.

Cristea, D.; Ide, N.; Romary, L. (1998): Veins Theory. An Approach to Global Cohesion and Coherence. In the *Proceedings of Coling/ACL.*

Grosz, B. and Sidner, C. (1986). Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, Vol. 12, N. 3.

Jordan, M.P. (1992). An Integrated Three-Pronged Analysis of a Fund-Raising Letter. In W.C. Mann and S.A. Thompson (eds.), *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text*, pp. 171-226.

Kehler, A. (2002). *Coherence, Reference and the Theory of Grammar*. CSLI Publications.

Mann, W.C. and Thompson, S.A. (1987). *Rhetorical Structure Theory: A Theory of Text Organization*. Technical Report ISI/RS-87-190.

Marcu, D. (1997). *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. PhD Thesis, Department of Computer Science, University of Toronto.

Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization.* The MIT Press. Cambridge, Massachusetts.

O'Donnell, M. (1997). Variable-Length On-Line Document Generation. In the *Proceedings of the 6th European Workshop on Natural Language Generation*, Gerhard-Mercator University, Duisburg, Germany.

Pardo, T.A.S. (2005). *Métodos para Análise Discursiva Automática*. PhD Thesis. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. São Carlos-SP, June, 211p.

Pardo, T.A.S. and Nunes, M.G.V. (2003). *A Construção de um Corpus de Textos Científicos em Português do Brasil e sua Marcação Retórica.* Technical Report N. 212. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. São Carlos-SP, September, 26p.

Pardo, T.A.S. and Nunes, M.G.V. (2004). *Relações Retóricas e seus Marcadores Superficiais: Análise de um Corpus de Textos Científicos em Português do Brasil.* Technical Report N. 231. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. São Carlos-SP, April, 73p.

Pardo, T.A.S.; Nunes, M.G.V.; Rino, L.H.M. (2004). DiZer: An Automatic Discourse Analyzer for Brazilian Portuguese. In the *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence* – SBIA (Lecture Notes in Artificial Intelligence 3171), pp. 224-234. São Luis-MA, Brazil. September, 29 - October, 1.

Pardo, T.A.S. and Seno, E.M.R. (2005). Rhetalho: um corpus de referência anotado retoricamente. In *Anais do V Encontro de Corpora*. São Carlos-SP, November 24-25.

Pereira, F.C.N. and Warren, D.H.D. (1980). Definite Clause Grammars for Language Analysis – A Survey of the Formalism and Comparison with Augmented Transition Networks. *Artificial Intelligence*, N. 13, pp. 231-278.

Schauer, H. (2000). Referential Structure and Coherence Structure. In the *Proceedings of TALN*. Lausanne, Switzerland.

Soricut, R. and Marcu, D. (2003). Sentence Level Discourse Parsing using Syntactic and Lexical Information. In the *Proceedings of HLT/NAACL.*

Sumita, K.; Ono, K.; Chino, T.; Ukita, T.; Amano, S. (1992). A discourse structure analyzer for Japonese text. In the *Proceedings of the International Conference on Fifth Generation Computer Systems*, Vol. 2, pp. 1133-1140. Tokyo, Japan.

Classroom Lecture Recognition

Isabel Trancoso, Ricardo Nunes, and Luís Neves

INESC ID / IST, R. Alves Redol, 9,
1000-029 Lisbon, Portugal
Isabel.Trancoso@inesc-id.pt
http://www.l2f.inesc-id.pt/~imt/

Abstract. The main goal of this work is to provide automatic transcriptions of classroom lectures for e-learning and e-inclusion applications. The first experiments using a recognition system trained for Broadcast News resulted in word error rates near 60%, clearly confirming the need for adaptation to the specific topic of the lectures, on one hand, and for better strategies for handling spontaneous speech. This paper describes the different domain adaptation steps that lowered the error rate to 45%, with very little transcribed adaptation material. It also includes a qualitative analysis of the different types of error, focusing on the ones related to a very high rate of disfluencies.

1 Introduction

The goal of the national project LECTRA is the production of multimedia lecture contents for e-learning applications. Nowadays, the availability on the web of text materials from University courses is an increasingly more frequent situation, namely in technical courses. Video recording of classes for distance learning is also a more and more frequent possibility. Our contribution to these contents (text books, slides, exercises, videos, etc.) will be to add, for each recorded video, the synchronized lecture transcription. We believe that this synchronized transcription may be specially important for hearing-impaired students.

This project encompasses 5 tasks. The first one concerns the collection of the training and test material (both in terms of recorded audio-video signals and textual data) related to a selected course. In the second task, this training data is used to adapt the acoustic, lexical and language models of our large vocabulary continuous speech recognizer (optimized for broadcast news transcription) to the course domain, thus yielding the automatic transcription of the lecture contents.

From a research point of view, the lecture transcription domain is very challenging, mainly due to the fact that we are dealing with spontaneous speech (mostly from the same speaker). Spontaneous speech is characterized by strong coarticulation effects, non-grammatical constructions, hesitations, repetitions, filled pauses, etc. [1]. For e-learning purposes, a plain transcription may not be intelligible enough, and may need ”enrichment” with punctuation, capitalization, marking of disfluencies, etc. These research challenges are the focus of the third task of the project.

The two last tasks of this project deal with integration and user evaluation, respectively. The goal is to integrate the recorded audio-video signal and the corresponding

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 190–199, 2006.

transcription with the other multimedia contents and synchronize them according to topic, so that a student may browse through the contents, seeing a viewgraph, the corresponding part in the text book, and the audio-video signal with the corresponding lecture transcription as caption.

For the user evaluation we intend to use both normal hearing and hearing impaired students, evaluating the lecture transcription with and without manual correction. This latter evaluation will give us an indication of how close we are in terms of automatic lecture transcription to be able to use such tools in real-time in a classroom.

Lecture transcription has been the target of much bigger research projects such as the Japanese project described in [2], the European project CHIL (Computers In The Human Communication Loop) [3], and the American iCampus Spoken Lecture Processing project [4]. In some of these projects, the concept of lecture is different. Our classroom lectures are almost 90 minutes long, and they involve mostly a single speaker (the teacher) who tried to create a very informal atmosphere. This contrasts with the 20 minute seminars used in [3], where a more prepared speech can often be found. Unfortunately, the amount of material for adapting our recognizer to the lecture domain is also very different from the very large amounts collected in other projects.

Section 2 summarizes the first task of the project - corpus collection, which started with two very different courses. Section 3 describes our baseline recognizer and the corresponding results. Section 4 is dedicated to the adaptation of the recognizer modules to the domain of the 2 courses. Section 5 summarizes our preliminary efforts in the third task in terms of dealing with the different sources of error we have encountered. Our last Section will discuss future research plans.

2 Corpora Collection

Two very different courses have been selected for our pilot study: one entitled ”Economic Theory I” (ETI) and another one entitled ”Production of Multimedia Contents” (PMC). Both were taught during one semester. The ETI course and the first 7 classes of the PMC course were recorded with a lapel microphone. The last part of the PMC course was recorded with a head-mounted microphone.

The two recording types presented specific problems. The lapel microphone proved inadequate for this type of recordings given the very high frequency of head turning of the teacher (towards the screen or the white board) that caused very audible intensity fluctuations. The use of the head-mounted microphone clearly improved the audio quality. However, the wireless communication system between the microphone and the sound recorder involved an automatic gain control, which actively increased the gain during the students questions, due to their distance from the microphone. As a result, the following reply from the teacher was highly saturated. Overall, 11% of the recorded segments with the head-mounted microphone were saturated.

No attempt was made to record the participation of the students in the class. The teachers were motivated to repeat their questions before answering them, but this was not frequently done.

The audio signal was extracted from the wmv (Windows Media Video 9 Codec) recordings using the ”VideotoAudioConverter” software, and converted to wav format at 16 kHz sampling rate, 16 bits per sample.

Table 1. Duration and number of words in each manually transcribed set

	ETI			*PMC*		
	Train.	*Dev.*	*Test*	*Train.*	*Dev.*	*Test*
Duration (minutes)	62	46	36	73	52	42
Number of words	8k	7k	5k	10k	8k	6k

The recordings had variable duration, ranging from 40 to 90 minutes. All the ETI classes were taught by the same teacher, a male speaker with a Lisbon accent. Three of the PMC classes were taught by invited experts and the remaining 17 by the teacher, also a male speaker with Lisbon accent.

Due to very limited human resources, the manual transcription of all the classes of the two pilot corpora was totally infeasible. From the ETI course, we selected 3 segments of different classes, recorded several weeks apart, that served as training, development, and test sets. From the PMC course, we selected 3 other segments of distinct classes recorded by the main teacher, using the head-mounted microphone. The Transcriber software[1] was used for manually transcribing these segments. Table 1 shows the duration of the different sets for the two corpora and the number of words in each.

The very informal atmosphere of the PMC classes leading to highly spontaneous speech (even including laughter now and then) was not the only research challenge of this particular course. Because of its contents, it involved much computer jargon, usually derived from English (e.g. *email*, *software*), and a heavy use of spelt or partially spelt acronyms (e.g. *http*, *jpeg*). The computer jargon was generally pronounced very close to their English pronunciation, even introducing xenophones that are not part of the phone inventory for European Portuguese [5]. The percentage of technical terms in English in the PMC test corpus was 2.1%, a fact that will affect the lexical model, as we shall see below.

The ETI course also included some technical terms in English (e.g. *consumer price index*), but much more infrequently. The reference to mathematical variables and expressions, on the other hand, was much more frequent (e.g. $P1'$).

In order to adapt the language models to the domain of each course, we tried to get additional course materials (textbooks, viewgraphs, student reports, exam questions, etc.) which were first converted to text format and later processed by our text normalizer to expand abbreviations (KB - kilobyte(s), MHz - megahertz), numerals, etc. Finally, sentence boundary tags were added (<s>and </s >).

For the ETI course, we had a textbook and viewgraphs. Given the extension of the text book (452k words), we initially discarded the viewgraphs. For the PMC course, the textbook was in English, a frequent scenario in undergraduate engineering courses in Portugal. So, in order to train language models in Portuguese we only had viewgraphs, exam questions and student reports. The text included in the viewgraphs amounted to 25k words, the exam questions to 2k words, and the student reports to 23k words.

Viewgraphs are typically characterized by specific grammatical constructions which clearly differentiates this material from other textual sources. By analyzing a small set of sentences from the PMC viewgraphs (around 2k words), the percentage of verbs

[1] http://trans.sourceforge.net

that was obtained (9.1%) was much smaller then the one observed in a similar set of sentences from PMC reports (17.0%). The percentage of nouns, on the other hand, was much higher (42.2% in viewgraphs vs. 27.1% in reports). This different construction will have an obvious negative impact on the domain adaptation.

3 Baseline Recognizer

Our baseline large vocabulary recognizer was trained for Broadcast News (BN) in European Portuguese [6]. It uses hybrid acoustic models that try to combine the temporal modeling capabilities of hidden Markov models with the pattern classification capabilities of MLPs (Multi-Layer Perceptrons). The models have a topology where context-independent phone posterior probabilities are estimated by three MLPs given the acoustic parameters at each frame. The streams of probabilities are then combined using an appropriate algorithm [7]. The MLPs were trained with different feature extraction methods: PLP (Perceptual Linear Prediction), Log-RASTA (log-RelAtive SpecTrAl) and MSG (Modulation SpectroGram). Each MLP classifier incorporates local acoustic context via an input window of 7 frames. The resulting network has a non-linear hidden layer with over 1000 units and 40 softmax output units (38 phones plus silence and breath noises). The language model was created by interpolating a newspaper text language model built from over 400M words with a backoff trigram model using absolute discounting, based on the training set transcriptions of our BN database (45h). The perplexity (PP) is 139.5. The vocabulary includes around 57k words. For the BN development test set corpus, the out-of-vocabulary (OOV) word rate is 1.4%. The lexicon includes multiple pronunciations, totaling 66k entries.

For Broadcast News, this baseline recognizer achieves an average WER (word error rate) for all conditions of 32%, which decreases to 13%, in F0 conditions (read speech, studio recordings). These results were obtained at 4.4x real time in a Pentium IV Processor at 2.66GHz.

3.1 Audio Segmentation

The BN recognizer has a pre-processing module that performs audio segmentation [7]. This module segments the audio file into homogeneous regions according to background conditions, speaker gender and special speaker id (anchors). The speech/non-speech classification is used in the lecture transcription to exclude from the recognition task the segments containing questions or comments from the students. These segments are recorded with a very distant microphone, which makes them almost unintelligible in many cases. Besides excluding almost all these segments, the pre-processing module also excludes some very noisy segments with the teacher's voice.

We had about 170s with student contributions in each of the two test sets. The classification module excluded almost all of these segments (except for 9s in the ETI test set and 3s in the PMC test set).

3.2 Recognition Results

The BN recognizer described above was first applied to the transcribable segments, without any adaptation. The WER was 56.4% and 63.6%, respectively, for the ETI and

PMC test sets. These very bad results were expected in view of the fact that we are dealing with spontaneous speech recorded in a classroom, with very specific contents. Furthermore we had to cope with recording problems related either to head shifts relative to microphone positioning in one case, or to very frequent saturation effects in another. The lack of domain adaptation is specially patent in the high OOV rates and perplexity values obtained for the PMC test set (OOV=3.4%, PP=292.8). For the ETI test set, the values were much lower (OOV=1.6%, PP=175.0).

4 Domain Adaptation

The following subsections describe the adaptation stages to the lecture domain of the lexical, language and acoustic models. We tried to make this process as automatic as possible in order to enable the rapid porting to other course domains.

4.1 Lexical Model

For one of the two courses selected for our pilot study (PMC), the simplest approach of adding new entries to the pronunciation lexicon by running them through an automatic grapheme-to-phone conversion module for European Portuguese [8] would not be advisable, given the high percentage of technical terms of English origin.

Our first attempt consisted of designing a set of hand-crafted rules to separate the words that would likely be of foreign origin. The regular expressions dealing with grapheme sequences were written using the flex program and achieved a correct identification rate of 65% on the PMC viewgraphs training set. Given the high miss rate, we tried the intersection with an English lexicon of approximately 118k entries, and a total of 127k multiple pronunciations.[2]

This procedure was followed by some phone mapping, from the English phone inventory to the European Portuguese one (using SAMPA). At this stage we have excluded the use of xenophones. In the context of lectures for undergraduate university students, foreign technical terms are most often pronounced fairly close to their English pronunciation (e.g. "position" would be pronounced as [pɐziʃɐn] or [poziʃɐn] instead of the Portuguese pronunciation [pozitjõ]). The variability caused by the possible degrees of nativization is still enormous [9]. The phone mapping between the English phone inventory and the Portuguese one may not be unique. As an illustration, take the English phone [θ], a xenophone which can be pronounced either as [s] (closest symbol in terms of pronunciation) or [t] (closest symbol in terms of orthography, since *th* sequences never occur in Portuguese).

The new PMC vocabulary includes around 3k new entries, of which 80.4% correspond to technical terms in English or acronyms. Like foreign words, acronyms are a special class that does not follow the common lexicon rules. The rules for spelling acronyms are trivial, but partially spelt or read acronyms in European Portuguese have much more complex rules and are characterized by a high degree of pronunciation variability, even among native speakers. Nowadays, we can also find many acronyms in the mention to URLs.

[2] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

For the ETI course, we have selected for the vocabulary all the words of the transcribed training material, plus the words of the text book that occurred more than 5 times. The new ETI vocabulary includes around 325 new entries, of which 17.5% correspond to acronyms. The most frequent OOVs of the ETI test set are references to mathematical variables (33.8%).

4.2 Language Model

Given the scarcity and inadequacy of written training material for the PMC pilot course, building a language model on that basis alone would give rise to a very high perplexity (256.1) and OOV rate (8.0%). The best results were hence obtained by interpolating the new language model with the one derived from the broadcast news domain. The new 3-gram language model was built using the SRILM toolkit [10], with modified Knesser-Ney discounting. Before interpolation, the WER corresponding to this new model was 64.8%. After interpolation, it decreased to 58.7%. The perplexity decreased to 208.6 and the OOV rate to 1.7%.

For the ETI course, the interpolation of the textbook and the transcribed training lecture with the BN model decreased the WER to 54.3%, corresponding to a perplexity of 127.7. The OOV rate was practically the same as with the initial BN model.

4.3 Acoustic Model

Although the transcribed training material was also very scarce, it was worth testing how much one could gain from adapting the acoustic models to the speaker and classroom environment, with just one lecture. The adaptation procedure was slightly modified to take into account that we had no initial models for filled pauses. We tried adapting the acoustic models without and with language model adaptation. In the first tests, after 3 iterations, the WER was down to 48.0%, for the PMC course and to 45.4% for the ETI course. The WER reduction was hence much more significant than with language model adaptation alone. In the second tests, the WER decreased to 44.8% for the PMC course, and to 44.7% for the ETI course.

5 Error Analysis

A clear course for improving the word error rate is to get more training data, namely in terms of additional transcribed material. Another course is to make a qualitative and quantitative error analysis, hoping that our small test set would be representative enough to indicate typical error sources. This section is our first step in this direction. The qualitative analysis of the errors for the two test sets indicates the following types of error:

- Errors due to severe vowel reduction. Vowel reduction, including quality change, devoicing and deletion, is specially important for European Portuguese, being one of the features that distinguishes it from Brazilian Portuguese and that makes it more difficult to learn for a foreign speaker. It may take the form of (1) intra-word vowel devoicing; (2) voicing assimilation; and (3) vowel and consonant deletion and coalescence. Both (2) and (3) may occur within and across word boundaries.

Contractions are very common, with both partial or full syllable truncation and vowel coalescence. As a result of vowel deletion, rather complex consonant clusters can be formed across word boundaries. Even simple cases, such as the coalescence of the two plosives (e.g. *que conhecem*, 'who know'), raise interesting problems of whether they may be adequately modeled by a single acoustic model for the plosive. This type of error is not specific of spontaneous speech, being strongly affected by factors such as high speech rate. The relatively high deletion rate may be partly attributed to severe vowel reduction. It is also worth noting that 61.4% of the deleted words in the PMC test set are (typically short) function words.

- Errors in inflected forms. This affects mostly verbal forms (Portuguese verbs typically have above 50 different forms, excluding clitics), and gender and number distinctions in names and adjectives. It is worth exploring the possibility of using some post-processing parsing step for detecting and hopefully correcting some of these agreement errors. Some of these errors are due to the fact that the correct inflected forms are not included in the lexicon. It is known that one OOV term can lead to between 1.6 and 2 additional errors [11]. 32.0% of the OOVs in the PMC test set are verbal forms. The current lexicon does not have too many verbal forms with clitics (e.g. *desenvolveu-se*, 'developped'). It may be worth exploring the possibility of separating clitics when building the lexical and language models, although our previous attempts of doing some morphological analysis have not yet brought any significant improvements [12].
- Errors around speech disfluencies. This is the type of error that is most specific of the spontaneous speech of our lecture corpus. The frequency of repetitions, repairs, restarts and filled pauses is very high, in agreement with values of one disfluency every 20 words cited in [1]. Unfortunately, the training corpus for Broadcast News included a very small representation of such examples, and our manually transcribed corpus was far too small.
- Errors in tag questions. This type of construction is fairly frequent in both courses, given the need felt by the teachers to make sure that the class was following their presentations. Therefore the teachers often invited the students to give feedback by using tag questions such as *no ?* ('isn't it?', 36 instances in the ETI test set), *(es)t bem?* ('all right?'), *(es)to a ver?* ('are you understanding?'), or the nativized version of *okay?*. The very casual articulation of these words, coupled with the virtual non-representiveness of such examples in the written corpus makes them very difficult to be recognized.

The type of errors around disfluencies is the one that mostly differentiates this corpus from other corpora we have worked on, and therefore deserves our particular attention. At this early stage, it was not yet possible to make a quantitative analysis of all type of fillers [13] found in our corpus: filled pauses, discourse markers, explicit editing terms and asides/parentheticals. So far, we have mostly concentrated on filled pauses which are the easiest ones to automatically detect.

Filled pauses are hesitation sounds that speakers employ to indicate uncertainty or to maintain control of a conversation while thinking of what to say next. They can occur anywhere in the stream of speech. For European Portuguese, following the proposal of [14], the most common filled pauses were transcribed as *mm* (when the sound is produced with a closed mouth, sounding either as [ɨ̃ː] or [ũː]), *aam* (when there is evidence either of a nasal vowel or a vowel followed by a nasal murmur, sounding as [ɨ̃ː], [ɐ̃] or [ɐ:m]), and *aa* (when the sound corresponds to a non-nasal vowel, similar to either [ɨː] or [ɐ:]). The most common filled pause in our corpus of lectures is by far *aa*, very similar to the Portuguese article and preposition *a*, one of the top 5 most frequent words.

In the PMC test corpus, 1.9% of all manually transcribed words were filled pauses. In the ETI test corpus, the percentage was much lower (0.4%), although it had higher values in the training and development sets.

For filled pause detection, we implemented a method based on the relative stationarity of F0 and spectral slope over filled pauses. This method, which was far simpler but not as efficient as the one described in [15], has a 35.2% recall rate (number of filled pauses detected correctly / total number of filled pauses) and a 99.9% precision rate (number of filled pauses detected correctly / total number of filled pauses detected). This was the first step towards building acoustic models for filled pauses, which were not previously included in our BN recognizer.

Repetitions are also relatively easy to detect, when the repeated material is exactly the same as originally pronounced (29 instances of one-word and two-word repetitions in the TEI test set and 22 for PMC), but more complex revisions are more frequent.

Discourse markers are also extremely frequent in our corpus. The most typical discourse marker is by far *portanto* ('so'), which was pronounced in many different ways, mostly in very reduced forms, in both courses (104 instances in the ETI test set and 42 in the PMC test set). In fact, the recognition error of this particular discourse marker was very high (71.4% in the ETI test set and 60.2% in the PMC test set). The nativized version of *okay* was also fairly frequent in the PMC course. In the ETI course, on the other hand, we could find many other examples of discourse markers, such as *ora bem* ('well', 7 instances, 57.1% error rate), and *reparem* ('notice', 26 instances, 69.2% error rate). This variability in using discourse markers is very speaker and dialect dependent.

The individual discourse style differences are also very interesting. For instance, the ETI teacher uses rethorical (or hypothetical) questions very often, whereas the PMC teacher prefers statements or questions to the audience.

A significant part of the recognition errors occurs for function words. In the ETI test set, 44.5% of all words are function words, and the percentage is similar for the PMC test set (45.0%). 47.3% of all recognition errors in the ETI test set occur for function words, and 42.9% in the PMC test set. These results make us believe that the current performance, although too bad in terms of transcription, may be good enough for indexation purposes. This is specially important for the lecture browsing application, but this feature has not yet been included.

Error bursts, i.e. sequences of wrongly recognized words, were fairly frequent (e.g. around disfluencies). In the PMC test set, only 19.2% of the errors occurred in isolation; 18.6% occurred in bursts of two errors; 53.2% in bursts of 3-9 errors; and 8.9% in longer bursts. Similar statistics could be found for the ETI test set.

In the above analysis, we have not mentioned errors due to inconsistent spelling of the manual transcriptions, which were, however, relatively frequent. The most common inconsistency consists of writing the same entries both as separate words and as a single word (e.g. colormap and color map).

6 Conclusions and Future Work

This pilot study with lecture transcription allowed us to learn valuable lessons in terms of recording protocols, and validated the well known importance of large quantities of textual and manually transcribed material for training language and acoustic models. Despite the very limited resources, our domain adaptation efforts yielded a significant (although not sufficient) word error rate reduction.

Further error reductions must be obtained at the cost of better strategies for dealing with disfluencies. However, some of the identified error sources are not exclusive to spontaneous speech recognition. In fact, we are currently dealing with them in the scope of our efforts for automatic captioning of broadcast news. We believe that the use of much larger speech and text corpora may obviously decrease these problems, namely by using context-dependent acoustic models, but much can be gained by studying phenomena such as vowel reduction.

Producing a rich transcription for lectures does not only entail dealing with disfluencies, but also punctuation and capitalization, which are the focus of another PhD thesis in the scope of this project. The research challenges are enormous, not only in terms of disfluency detection and repair [16] [17] [18], but also in terms of producing a surface rich transcription [19] that is more intelligible for hearing impaired students.

New corpora have already been recorded in the current semester, not only for e-learning purposes, but most specially for helping an undergraduate student with progressive hearing disabilities. We are currently dealing with the additional challenges posed by a course on Algebra, which is particularly interesting, as it involves mentioning mathematical variables and expressions.

It will be worth investigating whether some time savings in terms of speaker adaptation can be achieved by asking the teacher to record a (read) technical text prior to starting the course. The manual transcription of the lectures themselves, however, remain very important for researching spontaneous speech phenomena, in particular speaker-dependent disfluencies [18]. It will also be worth investigating whether the existence of transcribed data for one course can be beneficial for another one, in a totally different domain. This would mean building a corpus of University lectures for several courses.

Acknowledgment. The authors would like to thank Ciro Martins, Hugo Meinedo, João Neto and Cu Viana for many helpful discussions. This work was partially funded by FCT project POSC/PLP/58697/2004. INESC-ID Lisboa had support from the POSI Program of the "Quadro Comunitrio de Apoio III".

References

1. Shriberg, E.: Spontaneous speech: How people really talk, and why engineers should care. In: Proc. Interspeech '2005, Lisbon, Portugal (2005)
2. Furui, S., Iwano, K., Hori, C., Shinozaki, T., Saito, Y., Tamura, S.: Ubiquitous speech processing. In: Proc. ICASSP '2001, Salt Lake City, USA (2001)
3. Lamel, L., Adda, G., Bilinski, E., Gauvain, J.L.: Transcribing lectures and seminars. In: Proc. Interspeech '2005, Lisbon, Portugal (2005)

4. Glass, J.R., Hazen, T.J., Hetherington, I.L., Wang, C.: Analysis and processing of lecture audio data: Preliminary investigations. In: Proc. Human Language Technology NAACL, Speech Indexing Workshop, Boston (2004)
5. Lindstrm, A.: English and Other Foreign Linguistic Elements in Spoken Swedish: Studies of Productive Processes and Their Modelling Using Finite-State Tools. PhD thesis, Linkping University (2004)
6. Trancoso, I., Neto, J., Meinedo, H., Amaral, R.: Evaluation of an alert system for selective dissemination of broadcast news. In: Proc. Eurospeech '2003, Geneva, Switzerland (2003)
7. Meinedo, H., Neto, J.: Audio segmentation, classification and clustering in a broadcast news task. In: Proc. ICASSP '2003, Hong Kong (2003)
8. Caseiro, D., Trancoso, I., Oliveira, L., Viana, C.: Grapheme-to-phone using finite state transducers. In: Proc. 2002 IEEE Workshop on Speech Synthesis, Santa Monica, CA, USA (2002)
9. Trancoso, I., Viana, C., Mascarenhas, M., Teixeira, C.: On deriving rules for nativised pronunciation in navigation queries. In: Proc. Eurospeech '1999, Budapest, Hungary (1999)
10. Stolcke, A.: Srlim - an extensible language modeling toolkit. In: Proc. ICSLP '2002, Denver, USA (2002)
11. Gauvain, J., Lamel, L., Adda, G.: Developments in continuous speech dictation using the arpa wsj task. In: Proc. ICASSP '1995, Detroit, USA (1995)
12. Martins, C., Neto, J., Almeida, L.: Using partial morphological analysis in language modeling estimation for large vocabulary portuguese speech recognition. In: Proc. Eurospeech '1999, Budapest, Hungary (1999)
13. LDC: Simple metadata annotation specification version 6.2. Technical report, Linguistic Data Consortium (2004)
14. Mata, A.: For a Study of Intonation in Spontaneous and Prepared Speech In European portuguese: Methodology, Results and Didactic Implications (in Portuguese). PhD thesis, FLUL, Lisbon (1998)
15. Goto, M., Itou, K., Hayamizu, S.: A real-time filled pause detection system for spontaneous speech recognition. In: Proc. Eurospeech '1999, Budapest, Hungary (1999)
16. Heeman, P., Allen, J.: Speech repairs, intonational phrases and discourse markers: modeling speakers' utterances in spoken dialog. Computational Linguistics **4**(25) (1999) 527–571
17. Johnson, M., Charniak, E.: A tag-based noisy channel model of speech repairs. In: Proc. ACL, Barcelona, Spain (2004)
18. Honal, M., Schultz, T.: Automatic disfluency removal on recognized spontaneous speech - rapid adaptation to speaker-dependent disfluencies. In: Proc. ICASSP '2005, Philadelphia, USA (2005)
19. Snover, M., Schwartz, R., Dorr, B., Makhoul, J.: Rt-s: Surface rich transcription scoring, methodology, and initial results. In: Proceedings of the Rich Transcription 2004 Workshop, Montreal, Canada (2004)

Semi-supervised Learning for Portuguese Noun Phrase Extraction

Ruy Milidiú[1], Cicero Santos[1], Julio Duarte[2], and Raúl Rentería[1]

[1] Departamento de Informática,
Pontifícia Universidade Católica, Rio de Janeiro, Brazil
milidiu@inf.puc-rio.br, nogueira@inf.puc-rio.br,
renteria@inf.puc-rio.br

[2] Centro Tecnológico do Exército, Rio de Janeiro, Brazil
jduarte@ctex.eb.br

Abstract. Semi-supervised learning is frequently used when we have a small labeled training set but a large set of unlabeled samples. In this paper, we combine Hidden Markov Models and Transformation Based Learning in a semi-supervised learning approach. Self-training and Co-training are the two semi-supervised techniques that we apply to our scheme in order to classify Portuguese noun phrases. Our main goal here is to show that we can achieve effective noun phrase extraction using fewer tagged examples by applying a semi-supervised technique. Our models show good improvement with a small labeled corpus and little with a large one.

1 Introduction

Supervised Learning is the best alternative when it comes to train a model. Although this technique is very efficient, it is also very expensive because it needs a set of labeled samples, which must be specifically build and usually require intensive human labor. Hence, to generate high quality examples is a laborious and expensive human task.

The main goal of semi-supervised learning is to take advantage of massive untagged samples that are a by-product of ordinary enterprise processes. Based on this large sample set, it is not hard to infer several statistical properties of the domain that can be very helpful on designing efficient training schemes.

In this work, two basic semi-supervised learning approaches, Self-training and Co-training, are used for Noun Phrase extraction in Portuguese corpora which is a very important task in various applications of Natural Language Process (NLP), Information Extraction (IE) and Information Retrieval (IR). NP extraction for the Portuguese was recently investigated by Miorelli [1], which used a Left-to-right Rightmost (LR) syntax analyzer with a NP grammar. Santos [2] applied Transformation-Based Learning (TBL) to perform this task. Pierce and Cardie [3] introduced semi-supervised approaches for noun phrase extraction of the English Language.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 200–203, 2006.

The aim of this work is to improve the performance of the Portuguese NP extraction using semi-supervised techniques in conjunction with widespread machine learning techniques like Hidden Markov Models and TBL.

In our experiments, using labeled samples extracted from the SNR-CLIC corpus described in [4], and unlabeled ones from the Mac-Morpho Corpus, we obtained a 88.21% best F-score.

2 Modeling

Classification Confidence For each learning technique in the semi-supervised approach requires a confidence estimation to rank the classified sentences.

For the HMM model, we used the *log* probabilities calculated by the *Viterbi* algorithm, which is used to determine the best state sequence in a HMM.

Since the TBL algorithm does not provide any estimate for the token classification confidence, we developed a straightforward one:

Initial Classification Confidence: based on token frequencies;

Rule confidence: using the equation $q(r) = \frac{g-d}{g+b}$, where g is the number of mistakes corrected by the rule r in the training corpus, b is the number of mistakes generated by r and d is the number of mistakes generated by r but corrected by other rules.

Using these two information items, we estimate the confidence of a token classified by a TBL model as follows: (1) If no rules were applied to the token, the confidence is the one of the initial classification; (2) If any rule was applied, the confidence is the one of the last rule applied to it.

Given the token confidence, an estimate for a sentence classification confidence is defined as the smallest confidence among its tokens.

Semi-supervised Approaches. One quick way to model a noun phrase extraction with HMM is to use the np-tags as the hidden states, the pos-tags as the emissions and normalized tag frequencies as the probabilities. This simple model is very inefficient since it has a small number of states and it does not take advantage of the inherent structure of the noun phrases neighborhood.

In HMM Self-training, to overcome this limitation we introduce new enhanced replacing states, that take advantage of the inherent structure of the NPs' neighborhood. The HMM emissions can be also enhanced by using a combination of pos-tags words for special cases like the pos *PREP* and *KC*.

The HMM model often makes some simple tagging mistakes. These mistakes can be easily corrected by a set of fixed rules that are applied after the classification process. These rules are obtained by a set of TBL rules and evaluated with the corpus. We keep only the rules that show a positive score.

In TBL Self-training, the same initial classification introduced by Santos [2] is used, where each word is assigned the np-tag that is most frequently associated with its pos-tag, except for prepositions, where a lexicalized approach is used. For the template set, the one that shows the best results in the experiments in [2] is used. This set contains 86 templates and uses the features word, pos-tag and np-tag.

HMM Co-training uses two models, the first is the same used in our Self-training approach. The second differs only on the emission choice, where only words are considered. The final classifier is the the one that uses the pos-tags.

The TBL-HMM Co-training uses two different models: HMM Self-training and TBL Self-training.

3 Experimental Results

For each semi-supervised approach, two main tests are presented: one with a small corpus, and another with a large corpus. The small corpus contains 2, 246 tagged sentences and 2, 247 untagged sentences. The large corpus contains 4, 493 tagged sentences and 12, 530 untagged sentences. The tagged data is divided into three subsets, 50% of the tagged sentences for training, 25% for validation and the remaining 25% for testing. We assume the traditional F-Score as our key statistics.

At each iteration, 10% of the sentences from the initial training data is extracted from the untagged corpus and classified. The top 20% sentences regarding the confidence level are incorporated into the training data. We stop training whenever no improvement is observed. We show here the results of a cross-validation iteration, tough the other ones are quiet similar.

Figure 1 shows the results obtained with HMM Self-training modeling. In the Small Corpus, the model shows an improvement of almost 0.5% in the validation data, although this improvement was not followed in the test data. In the Large Corpus, the improvements were small, but the test data shows the same behavior as the validation data.

Figure 2 shows the results for the experiments using the TBL Self-training approach. We can see better improvement only in small corpus. This small improvement is mainly due to the fact that the initial model, totally supervised, is already very good.

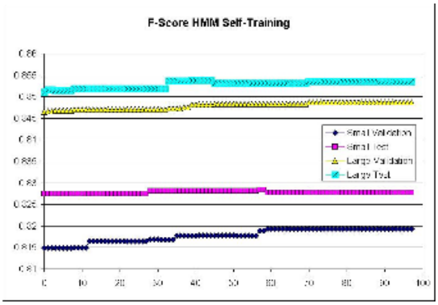


Fig. 1. F-Score for HMM Self-training

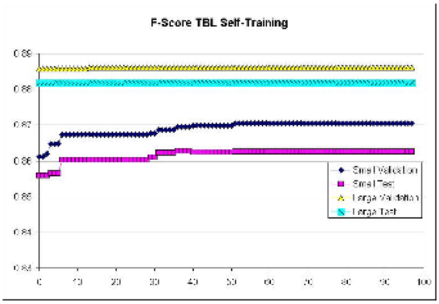


Fig. 2. F-Score for TBL Self-training

In Figure 3, HMM Co-training results are shown. We see better improvement, almost 1%, in the small corpus. Altough, very little improvement is noticed in large corpus.

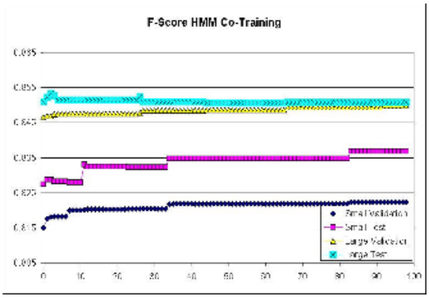


Fig. 3. F-Score for HMM Co-training

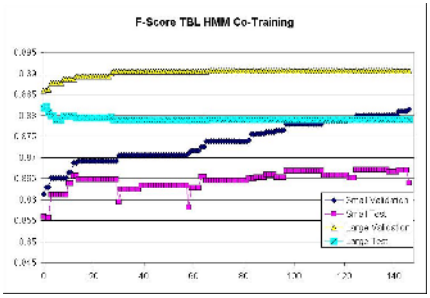


Fig. 4. F-Score for TBL-HMM Co-training

Finally, Figure 4 shows the results obtained by the TBL model with the TBL-HMM Co-training approach, where we can see good improvements in the validation data, although it is not followed in the test data.

4 Conclusions

The use of semi-supervised techniques can do a great improvement in machine learning approaches by avoiding the need of labeling a great amount of data.

In our experiments, little improvement is obtained when the models are already performing nicely. Nevertheless, unlabeled data can be used not only to check if the supervised model can be enhanced, but also to employ a smaller tagged corpus as the first totally supervised model.

Another contribution of this work is the proposal of a new way to quantify the classification confidence associated with a TBL classifier, based on the rules' score. The choice of states in HMM and templates in TBL can have a significant influence on the final results. We shall continue tuning these modeling elements in order to achieve even better results.

References

1. Miorelli, S.T.: Extração do sintagma nominal em sentenças em português. Master's thesis, Pontifícia Universidade Católica, Porto Alegre - RS (2001)
2. Santos, C.N.: Aprendizado de máquina na identificação de sintagmas nominais: o caso do português brasileiro. Master's thesis, IME, Rio de Janeiro - RJ (2005)
3. Pierce, D., Cardie, C.: Limitations of co-training for natural language learning from large datasets. In: Proceedings of the EMNLP-2001. (2001)
4. Freitas, M.C., Garrão, M., Oliveira, C., Santos, C.N., Silveira, M.: A anotação de um corpus para o aprendizado supervisionado de um modelo de sn. In: Proceedings of the III TIL / XXV Congresso da SBC, São Leopoldo - RS (2005)

Automatic Extraction of Keywords for the Portuguese Language

Maria Abadia Lacerda Dias[1] and Marcelo de Gomensoro Malheiros[2]

[1] UNICAMP - State University of Campinas, Campinas - SP, Brazil
abadialacerda@yahoo.com.br
[2] UNIVATES University Center, Lajeado - RS, Brazil
mgm@univates.br

Abstract. This paper outlines the adaptation of an algorithm for automatic extraction of keywords for the Portuguese Language. Keywords make possible to summarize the contents of documents in a compact form, and may also be used as an efficient measure of similarity between texts. This work is focused on the extraction of keywords for theses on several fields of knowledge. To identify the keywords the KEA algorithm was used, together with a stemming technique specific to Portuguese and a manually created list of stopwords. It is shown that the results obtained are good enough for practical use and similarly match what have been done for the English Language.

1 Introduction

The initial motivation for working with automatic extraction of keywords from texts came from the need of summarizing tools to be used with the UNICAMP Digital Library, to build a better search mechanism. The use of keywords may lead to powerful ways of filtering and organizing documents, therefore it is of major interest the development of methods to automatize keyword extraction.

This work describes the adaptation of the KEA algorithm [4] for the automatic extraction of keywords from texts in the Portuguese Language. To achieve this a specific stemmer for Portuguese was chosen, being described in Section 2. Next, in Section 3, it is described the construction of an adequate list of stopwords. In Section 4 a brief presentation of the KEA algorithm is made, and in the following section the adapted implementation is discussed, along with the result of an experiment made. In Section 6 the conclusion of this work is presented.

2 Stemming

Stemming is the process that combines the different forms of a word into a common representation, the **stem** (sometimes called radical) [3]. The stem is the resultant character set for the application of the stemming process to a given word. This is not necessarily equal to the linguistic root, but it is enough to deal with different variations of a word in the same way (for example, conector and conectores).

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 204–207, 2006.

The Porter stemmer is traditionally used in the literature, however it was developed specifically for the English Language. An adaptation of the Porter stemmer was made for Portuguese, however its overall precision showed to be limited, because it would frequently incur in overstemming errors (removing more characters than the necessary). Therefore this work used a modified implementation [2] of the Portuguese Stemmer algorithm, proposed in [3] specifically for the Portuguese Language.

3 List of Stopwords

Stopwords are frequent terms in a text which carry information of minor relevance. They are helpful in the task of selecting significant parts of phrases and identifying important words.

A research in literature did not result in an adequate set of stopwords, so it was decided to craft a new list, justifying the choice of each stopword to ensure confidence of the results.

The first step was to create a comprehensive list with the following classes of words: articles, pronouns, adverbs, prepositions, conjunctions, consonants, and vowels. Most of the words had been obtained from [1]. After the list was formed, it was noted that some words could also become nouns or adjectives, depending on the context. It was decided then to verify the usage of all words in the Houaiss dictionary, to clarify ambiguities. The words classified in more than one category had been kept in its more common classification. The words that had been classified primarily as adjective or nouns were excluded from the list.

The resultant list has 316 distinct words, including single letters (which occur frequently in texts automatically extracted from digital documents). The final list is available at [5].

4 The KEA Algorithm

KEA (Keyphrase Extraction Algorithm), presented in [4], is an algorithm to automatically extract keywords from texts in English. For this, it identifies candidate phrases using methods of lexical analysis, then calculates characteristic values for each candidate phrase and uses a Naïve Bayes Machine Learning technique for training and automatic extraction of keywords.

The machine learning technique constructs a prediction model using training documents with known keywords and then uses the constructed model to find keywords for new documents, that is, for documents whose keywords are not known. One way to evaluate the effectiveness of KEA is to analyze the extracted keywords against the document contents.

The KEA algorithm has two phases: training, when a statistical model is created using training documents with known author-assigned keywords; and extraction, when keywords are extracted from a new document using the model constructed previously.

During the phases of training and extraction, a set of candidate phrases is selected from the documents, which is the only step of KEA that depends on the document language. The algorithm considers all the word subsequences in each phrase and then determines which of these are adequate as candidate phrases. The criteria used is that candidates phrases cannot start or end with a stopword. Also, a stemmer is used to simplify the candidate phrases, thus reducing the number of candidates. Therefore, a specific list of stopwords and a specific stemmer are needed for each language to be used by the KEA algorithm.

The extraction phase makes possible to find automatically keywords for a new document, by selecting candidate phrases from this document and applying the knowledge stored in the model constructed before. The process is able to determine the probability of each candidate phrase to be a keyword, thus creating a ranked list. The candidates with higher probability are selected as keywords for the new document.

5 Implementation and Results

KEA has an implementation in the Java programming language, being distributed as Free Software [6]. The present work was based on the 3.0 version.

The KEA implementation has only two parts directly related to the language of the documents being processed: one that lists the set of stopwords and other that implements the stemming algorithm. As the official implementation contains support only for English, the modified stemmer cited in Section 2 and the list of stopwords described in Section 3 were added as new modules (available at [5]).

To evaluate the process of keyword extraction for texts in Portuguese, documents from many areas of knowledge were used. The training and extraction phases were then carried out several times, varying both the parameters for the

Table 1. Author classification of automatically extracted keywords; correct matches are shown in *italics*

model 1	model 2	model 3	model 4
pontos críticos	*primitivas*	*primitivas*	*primitivas*
primitivas	*pontos críticos*	*pontos críticos*	superfície
superfície	implícita	implícita	*superfície implícita*
superfície implícita	*superfície implícita*	*superfície implícita*	*conjuntos de nível*
regiões de influência	*regiões de influência*	*regiões de influência*	*pontos críticos*
conjuntos de nível	limiar	limiar	*regiões de influência*
função implícita	*conjuntos de nível*	*conjuntos de nível*	*função implícita*
malha	*topologia*	*topologia*	malha
modelo implícito	*modelo implícito*	*Implicit Surfaces*	*modelo implícito*
topologia	*função implícita*	duas primitivas	topológica
limiar	malha	esqueleto	esféricas
topológica	*algoritmo*	*modelo implícito*	*primitivas esféricas*
esféricas	configuração	*função implícita*	*suporte local*
críticos da função	topológica	malha	Luiz Henrique
numérica	Surfaces	configuração	Figueiredo

creation of models and the composition of the document sets. The complete results are described in [2].

One of the experiments made consisted in the inspection of the automatically generated keywords for three theses performed by their own authors. For each document, extractions were made using four models trained with 20 documents. For each situation, 15 keywords had been extracted and judged useful or not useful by the respective author. Table 1 shows the results for one particular thesis, where the words selected as representative are shown in *italics*.

In accordance with [4], author-assigned and automatically extracted keywords are sufficiently similar, but it is not difficult to guess which ones are from the authors. It can be verified in Table 1 that sometimes KEA chooses good keywords, but also chooses some that are improbable of being selected by authors, like the words topológica or esféricas. Despite these anomalies, the keywords extracted supplies an adequate description of the document. In the case where no keywords from the author are available, the choices from KEA could be a valuable resource to summarize or locate a document.

6 Conclusions

One of the contributions of this work is the adaptation of the KEA algorithm for the automatic extraction of keywords from documents written in Portuguese, evaluating its application in theses of different fields. Also, a list of stopwords was elaborated containing 316 words, whose careful construction was justified.

Acknowledgments

We would like to thank Prof. Dr. Mauro Sérgio Miskulin and Rubens Queiroz de Almeida, from UNICAMP, for their continued support and guidance.

References

1. Cunha, C.; Cintra, L. F. L.: Nova Gramática do Português Contemporâneo. 3 ed. Rio de Janeiro: Nova Fronteira, 2001.
2. Dias, M. A. L.: Automatic Extraction of Keywords for the Portuguese Language Applied to Theses in the Engineering Field. Master thesis (in Portuguese, to be published).
3. Orengo, V. M.; Huyck, C. R.: A Stemming Algorithim for The Portuguese Language. In: Proceedings of the SPIRE Conference. Laguna de San Raphael: [s.n.], 2001.
4. Witten I. H. et al.: KEA: Practical automatic keyphrase extraction. In: Proceedings of the Fourth ACM Conference on Digital Libraries. [S.l.]: [s.n.], 1999.
5. `http://ensino.univates.br/~mald/`
6. `http://www.nzdl.org/Kea/`

Semi-automatically Building Ontological Structures from Portuguese Written Texts

Túlio Lima Baségio and Vera Lúcia Strube de Lima

Pontifícia Universidade Católica do Rio Grande do Sul,
(PUCRS) - Programa de Pós-Graduação em Ciência da Computação,
Av. Ipiranga 6681, Porto Alegre - RS, Brazil
{tbasegio, vera}@inf.pucrs.br

Abstract. Building ontologies is a complex and time consuming process. In this paper we present an approach for semi-automatic acquisition of relevant concepts and taxonomic relations among these concepts from texts. Our approach is based on those found in the literature but it aims to be suitable for Brazilian Portuguese written texts. This study is supported by a tool, and the results from a case study of application are briefly presented.

1 Introduction

According to Blázquez et al. [2], there is no generalized and tested methodology for ontology building and different proposals of methodologies have been presented in the last years, however without playing the role of a standard methodology. Most of these proposals include manual ontology building, partially supported by computational tools. However, automatic or semi-automatic ontology acquisition would be extremely useful to many areas and it has brought researchers to consider, at a first moment, methodologies for semi-automatic ontology building [1, 3, 4, 5, 6], which start from knowledge found in texts of a specific domain. These approaches work with texts written in English, German or French. We aim at their application for semi-automatic acquisition of ontological structures from texts written in Brazilian Portuguese for the extraction of concepts and the extraction of taxonomic relations.

2 Related Work

Researchers have proposed approaches and methods to improve human productivity during the ontology building process. Buitelaar et al. [5] present an approach for the extraction or extension of ontologies from texts, integrating ontology engineering with linguistic analysis. Degeratu and Hatzivassiloglou [6] propose a method to build ontologies using information available in raw texts. Lame's method [4] aims at the identification of concepts through terms found in texts as well as the identification of semantic relations looking for the syntactic relations between terms. Maedche [1] presents a semi-automatic method to acquire a domain ontology, having a prior ontology as the basis for integration of new concepts and relations. Velardi et al. [3] use text mining techniques to build ontologies. Cláudia Pérez [7] presents a semi-

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 208 – 211, 2006.

automatic approach for knowledge extraction from texts in Brazilian Portuguese, which is used to construct Conceptual Maps. Two main points differentiate our research from Pérez's work. First, in the concepts acquisition phase, her work does not include a measure to decide on the selection of concepts. Second, her work selects predicative relations, i.e., those where the concepts are linked to verbs where they act as subject or object. Differently, our proposal is to identify the taxonomic relations between the concepts. Conceptual maps are more flexible, while the ontologies have classes, sub-classes and other well defined relations as, for example, meronymy and hypernymy. Hearst [8] presents lexical-syntactic standards for identification of the hyponymy and hypernymy relations from texts written in English, i.e., relations that arrange the concepts in a hierarchy. In [9], Morin and Jacquemin present standards for the acquisition of hypernymy relations in a French language corpus. Some standards in [9] are equivalent to the standards identified by Hearst in [8].

3 Proposal

From combined techniques found in the bibliography, we propose an approach for the identification of relevant concepts of the domain and the taxonomic relations among these concepts. The input to our approach is a linguistically annotated corpus including, for each term, besides its original form, lemma and grammatical label (pos-tag).

The identification of the relevant terms of the domain is done in six steps:

1. Eliminate through a stopword list the terms that do not represent concepts and eliminate the terms containing non-alphabetical characters, proper names and abbreviations;
2. Weigh the frequency of the remaining terms, using the combination of measures called TFIDF (term frequency x inverse document frequency);
3. Define a minimum frequency for a term to be considered relevant to that domain;
4. Manually exclude terms judged unnecessary or incorrect to the domain;
5. Identify compound terms from the list of relevant terms, selecting those containing at least one relevant term.
6. Identify the variants of terms, looking for sequences of two words formed by a noun (relevant term) and an adjective, regardless the order how they appear.

The second phase, composed by five steps, is to identify taxonomic relations among the relevant terms detected in the previous phase.

1. Identify taxonomic relations from the syntactic head of compounds terms, relating each multi-word term to the relevant term that is part of its composition.
2. Identify taxonomic relations using the variants of the terms. The idea is to connect each variant with the relevant term that is part of its composition.
3. Identify taxonomic relations according to the standards proposed by Hearst [8]. The idea is looking for Hearst standards where there is at least one relevant term.
4. Identify taxonomic relations in the texts through the standards proposed by Morin and Jacquemin [9]. For each standard there should be at least one relevant term.
5. Identify and exclude the duplicate relations extracted in the previous steps.

The next step is code generation for the ontological structure. In this phase, the goal is to use the acquired relevant terms and taxonomic relations, in order to generate an ontological structure in an ontological representation language.

4 Case Study

We used a document collection extracted from the NILC corpus composed by texts of the Jornal Folha de São Paulo, 1994, pos-tagged. The texts used are from the section of tourism, and this subset of the corpus contains 294 documents with 88601 words.

4.1 Identification of Terms

The first step resulted in a total of 60768 removed words, about 68,59% of the corpus. Through TFIDF measure, all 27833 remaining words, nouns in most of the cases, were weighed. The three main relevant terms according to TFIDF were: ilha (island), praia (beach) and hotel (hotel). Thus, we excluded all terms where the weight was below defined threshold. A total of 747 terms remained. The fourth step was skipped in the case study due its simplicity. From the 747 remaining terms, 98 compound terms which contained at least one relevant term had been identified. E.g., costa norte de a ilha (north coast of the island), praia de areia (sand beach), quarto de hotel (hotel room). The last step resulted in 92 variants of terms containing at least one relevant term in their composition. E.g., ilha virgem (virgin island), ilha grande (big island), praia fluvial (pluvial beach).

4.2 Identification of Taxonomic Relations

A limitation of the prototype used in these tests is that it doesn't consider compound terms and variants of the terms in the identification of Hearst's standards (step 3).

Table 1. Results from the identification of taxonomic relations

Step	Extracted relations	Example
1	98	*praia* → *praia de areia, praia da badalação*
2	92	*praia* → *praia fluvial, praia paulista, praia selvagem*
3	45	*artesanato* → *arcos, flechas, tacapes, cocares, tangas*
4	41	*armadilha* → *redes, fisgas, ganchos, garatéias e espinhéis*

5 Final Remarks

Based on methods and techniques found in the literature, we proposed an approach for automatic identification of relevant terms of the domain and taxonomic relations among these terms. Even if there are several semantic relations among lexemes and their meanings, in this work we choose to deal with taxonomic relations only, because ontologies are structured as a hierarchy of concepts (taxonomy). We developed a prototype to help ontologists with the main steps necessary to assist in the ontology building process. Our intention isn't only to implement a collection of existing methods for identification of concepts and relations among these concepts, but rather to come to an approach that enables a systematic regard to the task of ontology building. Besides, the objective of this proposal is to provide support to the ontology engineer, in a way that he doesn't need to be a specialist in the domain of the ontology being built. Another important point to be observed is that the techniques tested need manual validation. In our tests, the results obtained showed that this approach needs to be refined. The prototype is in use in order to identify these needs of refinement through more exhaustive tests. In order to evaluate the results obtained, we are preparing Kappa statistic tests, which are widely accepted in the field of content analysis. The future work will be oriented to conclude the case study and to make a detailed analysis of its results.

References

1. Maedche, A.: Ontology Learning for the Semantic Web. Kluwer Academic Publishers (2002).
2. Blázquez, M., Fernández, M., García-Pinar, J. M., Gómez-Pérez, A.: Building Ontologies at the Knowledge Level Using the Ontology Design Environment. In: Proc. of the Knowledge Acquisition Workshop, KAW98 (1998).
3. Velardi, P., Paolo, F., Missikoff, M.: Using text processing techniques to automatically enrich a domain ontology. In: Proc. of the International Conference on Formal Ontology in Information Systems – FOIS. Ogunquit, Maine, USA (2001) 270-284.
4. Lame, G.: Using text analysis techniques to identify legal ontologies components. In: Proc. of the Workshop on Legal Ontologies of the International Conference on Artificial Intelligence and Law (2003).
5. Buitelaar, P., Olejnik, D., Sintek, M.: A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In: Proc. of the European Semantic Web Symposium (ESWS), Heraklion, Greece, May (2004).
6. Degeratu, M., Hatzivassiloglou, V.: An Automatic Method for Constructing Domain-Specific Ontology Resources. In: Proc. of the Language Resources and Evaluation Conference (LREC), Lisbon, Portugal, May (2004).
7. Pérez, C.: Aquisição de conhecimento a partir de textos para a construção de mapas conceituais. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale do Rio dos Sinos, São Leopoldo (2004).
8. Hearst, M. A.: Automatic acquisition of hyponyms from large text corpora. In: Proc. of the 14th International Conference on Computational Linguistics, Nantes, France, July (1992) 539-545.
9. Morin, E., Jacquemin, C.: Automatic acquisition and expansion of hypernym links. Computer and the Humanities, Kluwer Academic Press, vol. 38, n.4 (2003).

On the Use of Machine Learning and Syllable Information in European Portuguese Grapheme-Phone Conversion

António Teixeira[1], Catarina Oliveira[2], and Lurdes Moutinho[2]

[1] Dep. Electrónica e Telec/IEETA, Universidade de Aveiro, Portugal
[2] Centro de Línguas e Culturas, Universidade de Aveiro, Portugal

Abstract. In this study evaluation of two self-learning methods (MBL and TBL) on European Portuguese grapheme-to-phone conversion is presented. Combinations (parallel and cascade) of the two systems were also tested. The usefulness of syllable information is also investigated. Systems with good performance were obtained both using a single self-learning method and combinations. Best performance was obtained with MBL and the parallel combination. The use of syllable information contributes to a better performance in all systems tested.

1 Introduction

This paper describes the development of EP g2p conversion modules based on machine learning methods. We investigated both the use of Memory Based Learning (MBL), Transformation Based Learning (TBL) and hybrid approaches.

Following recent results on the use of richer feature sets to improve machine learning systems, namely the use of syllable [1] and morphologic [2] information, we, also, tested the impact on systems' performance of using syllable information. This effort was possible due to the availability of an automatic syllabification procedure based on orthographic input [3].

2 EP g2p Using Two Machine Learning Methods

On the approach we adopted g2p conversion is a one-to-one mapping from a set of graphemes to a set of phones (Portuguese SAMPA). The phone set includes both the empty phone and phone clusters.

We selected two different machine learning methods for our experiments: Memory-Based Learning (MBL) and Transformation-Based Learning (TBL).

TBL was proposed by Brill, in 1995 [4]. Given a tagged training corpus, TBL produces a sequence of rules that serves as a model of the training data. To derive the appropriate tags, each rule may be applied in order to each instance in an untagged corpus. We selected the fnTBL tool, a customizable, portable and free source machine-learning toolkit [5].

MBL is based on the idea that intelligent behaviour can be obtained by analogical reasoning (see: [6]). Our MBL system is based on the use of TiMBL [6]. TiMBL implements several memory-based learning algorithms (IB1, IB2, IGTREE, TRIBL and TRIBL2).

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 212–215, 2006.

Features: The features, inspired in part by [2], on which our models were trained are: GRAPHEME - the current letter; POS_IN_SYLL - regarding position of the current grapheme within its syllable; SYLL_BOUND - specifying whether a syllable boundary follows or not; SYL_POS_WORD - position of the current grapheme syllable in the word; LEX_STRESS - information regarding stress position.

3 Evaluation Methodology

Metrics: Both **Word Error Rate (WER)** and **Phone Error Rate (PER)** have been chosen as evaluation measures. To complement these rates, we also used the **Mean Normalized Levenshtein Distance (MNLD)** of two strings as proposed in [2]. The mean of all normalized distances is calculated.

Corpora: The first train corpus was taken from the Portuguese version of Ispell and includes 6500 different entries. We started the creation of a new corpus consisting of the remaining Ispell entries. At the time of writing 4000 words were available, and combined with the original train corpus resulted in our second train corpus, of 10.5 kwords.

For evaluation, we created two test sets. The first, consists in 2076 common words, corresponding to a fraction of the Fundamental Portuguese corpus [7]. The second, consists of 1303 complex words randomly selected from the Publico corpus.

4 Experiments

We tested the 2 different approaches (MBL and TBL) still unexplored for our language; evaluated the usefulness of syllable information; evaluated the starting point for the TBL rules; and tested combinations of the basic systems.

Using MBL: As a first set of experiments we investigated the use of MBL and the influence of using syllable information on the results. Our tests with the different algorithms implemented by TiMBL pointed to a better performance of TRIBL2. We used the default configurations for each algorithm. We only present, in Table 1, the results using this algorithm.

In general, results for the 3 metrics improve when using syllable information and when using the bigger corpus for train. Values of PER and WER are better for test1 due to the smaller length of the words, being MNLD less sensitive to this difference in test corpora.

Table 1. MBL results using the TRIBL2 algorithm on the two training and test corpora

SYSTEM				test1			test2		
Num	Train	Syllable	Algor.	PER%	WER%	MNLD	PER%	WER%	MNLD
s1	6.5k	No	TRIBL2	5.01	27.26	0.056	6.68	44.43	0.063
s2		Yes	TRIBL2	3.88	22.06	0.045	5.67	37.51	0.051
s3	10.5k	No	TRIBL2	4.33	24.95	0.050	5.36	37.74	0.049
s4		Yes	TRIBL2	3.76	21.63	0.043	4.79	32.36	0.042

Table 2. TBL results on the two training, the two test corpora and the two 1st step alternatives

SYSTEM				test1			test2		
Num	Train	1st step	Syllable	PER%	WER%	MNLD	PER%	WER%	MNLD
s5	6.5k	table	no	7.90	43.00	0.088	8.66	56.07	0.091
s6			yes	5.09	27.73	0.057	5.15	36.48	0.055
s7		rules	no	5.42	29.23	0.061	5.94	38.33	0.063
s8			yes	4.85	26.96	0.055	4.65	33.18	0.051
s9	10.5k	table	no	6.71	37.22	0.077	7.04	48.23	0.076
s10			yes	4.26	23.74	0.049	4.03	29.34	0.043
s11		rules	no	4.58	25.13	0.053	5.01	33.56	0.055
s12			yes	4.19	23.74	0.049	3.89	28.42	0.043

Table 3. Results for the combination of our 2 data-driven methods with a rule based system

SYSTEM			test1			test2		
Num	Train	Syllable	PER%	WER%	MNLD	PER%	WER%	MNLD
s13	6.5k	no	3.36	19.16	0.038	6.53	44.44	0.061
s14		yes	2.77	16.23	0.032	3.13	22.91	0.027
s15	10.5k	no	2.79	16.42	0.032	5.30	38.20	0.049
s16		yes	2.66	15.75	0.031	2.91	21.37	0.025

Using TBL: For TBL we varied 2 things: the inclusion or not of syllable information in rules and the starting point for rule learning. For the latter, we used two alternatives: a simple table assigning the most common phone to each grapheme, or the result of an existent rule based system. Results are presented in Table 2.

Using syllable and a bigger corpus always results in a better performance according to the 3 metrics. Results are particularly bad for the systems using table lookup as first step and only grapheme information.

Using combinations: We tried 2 different combinations of our 2 basic systems (MBL and TBL) plus an existing rule-based system. The first consisted in exploring the parallel processing of each word by the 3 systems and keeping the decision of majority (Winner Take All method). The second consisted in exploring the different base idea of TBL, developed to create correction rules, and using the MBL as the first step for TBL. Results are presented in Tables 3 and 4.

Again results for the 3 metrics improve when using syllable information and a bigger training corpus. For the $6.5k$ training corpus the improvement when using syllable is particularly noticeable in test2 values of WER an MNLD, with a difference of -21.53 % and -0.034, respectively.

Results follow the tendency regarding syllable and size of training corpus and, in general, are very inferior to the WTA approach in all 3 metrics. Particularly bad are the results with only grapheme information. Clearly the TBL system is not able to correct the MBL errors when they are so many as in the grapheme-based system, and worst it seems to be contributing with additional errors. When using syllable information,

Table 4. Results for the cascade combination, using MBL as a first step for TBL

SYSTEM				test1			test2		
Num	Train MBL	Train TBL	Syllable	PER%	WER%	MNLD	PER%	WER%	MNLD
s17	6.5k	4k	no	17.16	60.38	0.178	17.93	74.50	0.185
s18			yes	3.83	21.86	0.044	4.71	33.56	0.049
s19	4k	6.5k	no	17.33	59.51	0.182	18.67	74.89	0.194
s20			yes	4.49	25.81	0.051	4.49	33.41	0.048

the MBL output is better but not enough for TBL to improve on its results. The MBL followed by TBL has worst results than the use of MBL alone.

Best results were obtained using the WTA method (WER of 15.75 % on test1 and MNLD equal to 0.025 on test2). These results compare favourably with the ones reported for German. For our best systems the 10 most common errors are all due to problems in the conversion of vowels (graphemes <e>,<o> and, less often, <a>) for test1, and the same vowels plus the <s> conversion to [S]/[Z] on test2.

5 Conclusion

This paper compares several self-learning approaches to EP g2p: MBL, TBL and their parallel and cascade combinations. Best results were obtained with the parallel combination of our 2 data-driven approaches with a rule-based system. The single system with overall better performance was the MBL. Results improved in general when using syllable information and a bigger training corpus.

We assume the limitations of our test results due to the utilization of two small and in-house developed corpora. The lack of a standardized test set for EP is a problem that we consider worth of attention in the future. With the ongoing work on the creation of a bigger training corpus, we expect to improve the results presented.

References

1. Marchand, Y., Damper, R.: Can syllabification improve pronunciation by analogy of english? Natural Language Engineering **1**(1) (2005) 1–25
2. Reichel, U.D., Schiel, F.: Using morphology and phoneme history to improve grapheme-to-phoneme conversion. In: Proc. InterSpeech Lisboa. (2005)
3. Oliveira, C., Moutinho, L.C., Teixeira, A.: On European Portuguese automatic syllabification. In: Proc. InterSpeech Lisboa. (2005)
4. Brill, E.: Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. Computational Linguistics **21** (1995) 543–565
5. Florian, R., Ngai, G.: Fast Transformation-Based Learning. (2001)
6. Daelemans, W., Zavrel, J., van der Sloot, K., van den Bosch, A.: TiMBL: Tilburg memory based learner, version 5.1, reference guide. Reference Guide ILK-0402, Tilburg University (2004) Reference: ILK Research Group Technical Report Series no. 04-02.
7. Nascimento, F., Marques, L., Segura, L.: Português fundamental: Métodos e documentos. Technical report, INIC-CLUL, Lisboa (1987)

A Model to Computational Speech Understanding

Daniel Nehme Müller, Mozart Lemos de Siqueira, and Philippe O.A. Navaux

The Federal University of Rio Grande do Sul,
Porto Alegre, Rio Grande do Sul, Brazil
{danielnm, mozart, navaux}@inf.ufrgs.br

Abstract. We propose a speech comprehension software architecture to represent the flow of the natural processing of auditory sentences. The computational implementation applies wavelets transforms to speech signal codification and data prosodic extraction, and connectionist models to syntactic parsing and prosodic-semantic mapping.

1 Introduction

This work argues that it is possible to unify several computational systems to represent the speech understanding process. Thus, we propose the SUM, a Speech Understanding Model, based on a neurocognitive model of auditory sentence (section 2). Through SUM, we search a computational representation for speech signal codification, prosody, syntactic and semantic analysis. The SUM is illustrated in the figure 1.

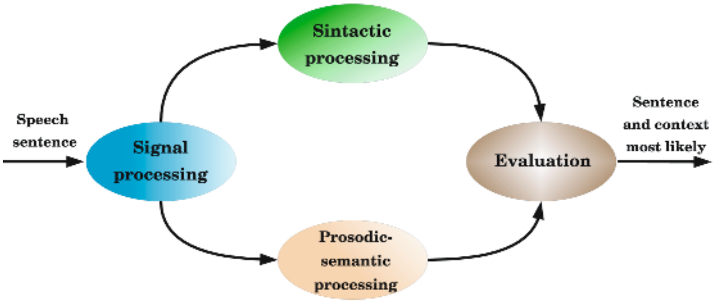


Fig. 1. The Speech Understanding Model - SUM

2 Neurocognitive Model

Angela Friederici [1] proposes a neurocognitive model of auditory sentence processing that identified which parts of the brain were activated at the time, given the different applied tests. She divided the processing of the auditory sentences in four large phases [1][2]. Indeed, the most recent research indicates that the

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 216–219, 2006.

prosody processing description must be added to the neurocognitive model [3]. In the first phase, it is done an acoustic characteristic extraction and codification. Thus, the prosodic characteristics, defined by the pitch variation, determine the processing segmentation. The linguistic characteristics will be analyzed at the syntactic level by the right hemisphere of the brain during the second phase [2]. The second phase performs the syntactic analysis and it occurs only in the left hemisphere of the brain. The semantic analysis is performed in the third phase and apparently awaits the syntactic analysis output in order to solve interpretation problems, brought about mainly by the words' categories contextualization. In the fourth and last phase the integration among syntax, semantics and prosody, necessary to revisit problems not resolved in the previous phases takes place. The syntax structure correction is necessary when there are lexical terms organization problems [2].

3 The Speech Understanding Model - SUM

From the four described phases in the neurocognitive model, we propose the architecture of SUM, illustrated in the figure 1. In SUM, the first phase extracts the coefficients from speech signal. These coefficients provide the information about the fundamental wave (F0) and they are used in the following phases. The second computational phase is the application of coefficients to realize the syntactic parsing. In the third phase the coefficients are used to semantic contexts definition. The fourth phase receives the analyses from second and third phases outputs. To each analyzed sentence the most likely context is indicated.

In the first phase of the computational model, the signal is processed by the application of wavelet transform. The second computational phase is the application of the wavelet coefficients to generation of temporal registers and parsing trees through the system SARDSRN-RAAM previously developed by Mayberry and Miikkulainen [4]. In the third phase, semantic and prosodic maps are applied using the Self-Organizing Map (SOM) [5]. The fourth computational phase performs the reception and the analysis of the output of the second and third phases. In this phase, the model indicates the most likely written sentence for a given speech sentence. The wavelet transform can be seen as a signal filter, making it possible to build filterbanks through them, and, thus, enabling a multiresolution analysis [6]. In this work we use the multiresolution analysis to speech signal codification. This process was split in phonetic and prosodic approaches. The phonetic approach is obtained from a single decomposition of wavelet coefficients (phonetic coefficients). The prosodic way is extracted from F0 variation (pitch). According to [7], to acquire information on the variations of the F0 speech it is necessary to detect the wavelet maximum points, which correspond to the glottal closure instants (GCI). If the maximum points are obtained, we attain the F0 estimation. The coefficients achieved (prosodic coefficients) will be sent to linguistic parsing system.

The syntactic analysis is allowed by the phonetic codification of words, extracted from wavelet transform, is structuralized through the RAAM, whose

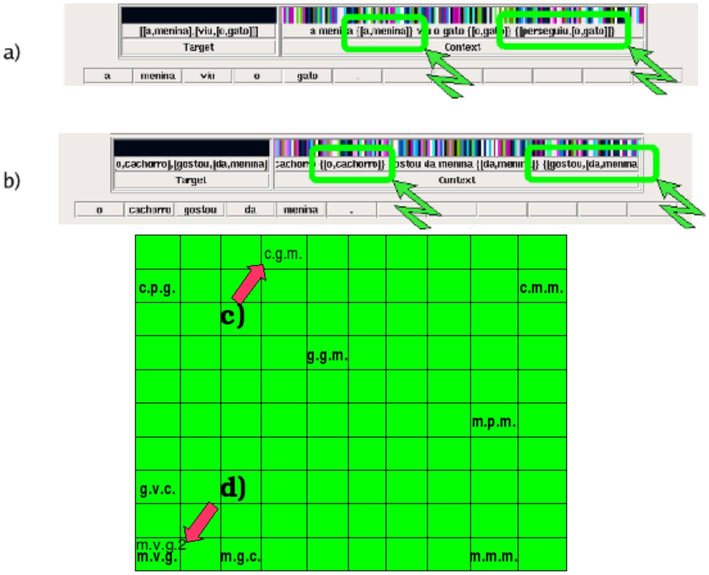

a)
[[a,menina],[viu,[o,gato]]]
Target
a meni a {[a,menina]} vi o gato {[o,ga]} {[perseguiu,[o,gato]]}
Context
a
menina
viu
o
gato
.
b)
o,cachorro],[gostou,[da,menina]
Target
cachor {[o,cachorro]} ostou da menina {[da,menin]} {[gostou,[da,menina
Context
o
cachorro
gostou
da
menina
.
c.g.m.
c.p.g.
c.m.m.
c)
g.g.m.
m.p.m.
g.v.c.
d)
m.v.g.2
m.v.g.
m.g.c.
m.m.m.

Fig. 2. Sentences recognition in SARDSRN and sentences map: the sentences a) and c) correspond to the c.g.m and b) and d) are the m.v.g.2

activation allows the sequencing of the words in the phrase by the SARDSRN-RAAM. Afterward, the temporal sequence of the component words is initiated, and the pattern presented in the input layer is distributed to the hidden layer and the SARDNET. This net, in turn, also feeds the hidden layer. Parallel to this hidden layer, there is a context layer, characterizing the SRN in the SARDSRN-RAAM. Finally, the output layer generates a pattern sentence that is decoded by the RAAM net. A relevant characteristic of the SARDSRN-RAAM is its great capacity to generate parsing sequences that will allow recognition of multiple parsing trees compressed in RAAM net. The semantic processing is composed by four chained SOM nets. In the first SOM net, the *prosodic map* groups the words according to signals derived from the analysis of variations in the F0 (prosodic coefficients). In the second SOM net, the *phonetic map* is obtained from the relations of phonetic characteristics of each word, extracted by wavelet transform. The net that forms the *semantic map* uses the output information on the activated neuron in the phonetic map plus the activation in the prosodic map. Finally, the last map is responsible for grouping sentences that are informed by the user. The composition of the output of semantic map for each word is the input of the *sentences map*. The recognition of speech patterns is performed by the sentences map, which indicates the most likely sentence. After syntactic and semantic processing, the systems' output are evaluated. The SARDSRN-RAAM system indicates an error rate (>=0.5) and the semantic maps system points to the winner neuron in the sentences map. If the syntactic processing has a high rate, we can do an approximation by semantic processing, and vice versa.

As illustration of the functionality of the system, two speech sentences not trained had been elaborated: m.v.g.2 - *a menina viu o gato* (the girl saw the cat) and c.g.m. - *o cachorro gostou da menina* (the dog liked the girl). The sentence m.v.g.2, to be presented to the syntax subsystem, resulted in the trained sentence *a menina perseguiu o gato* (the girl chased the cat) as an answer, thus pointing the error of the recognition (fig. 2a). In the sentences map, the identical positioning to the trained sentence m.v.g. - *o menino viu o gato* (the boy saw the cat) was obtained (fig. 2d). In the sentence c.g.m, the great distance (>2) from trained patterns in the sentence map indicates failure in recognition (fig. 2c). On the other hand, the syntactic system returned the exact written sentence, although it had *not* been trained in it (fig. 2b). These two examples mean that the first sentence corresponded to sentence that had more phonetic representations in common in the trained construction, and in the second sentence the system did not guarantee the semantic recognition, but would identify in syntactic system.

4 Conclusion

The resultant codification demonstrates that there is an interface between existent linguistic parsing connexionists systems to text analysis and the speech. This opens a new method to implementation of systems for written language with speech as input. The use of artificial neural nets in the syntactic and prosodic-semantic processing was presented as a facilitator in the language modeling process. The computational prototype, that demonstrates the processing of the SUM, resulted in a system of analysis by compensation. Therefore, when the syntactic analysis does not offer a good reliable level, it is possible to evaluate prosodic-semantic analysis, such as in human speech understanding.

References

1. Angela D. Friederici, "Towards a neural basis of auditory sentence processing," *Trends in Cognitive Sciences*, vol. 6, pp. 78–84, 2002.
2. Angela D. Friederici and Kai Alter, "Lateralization of auditory language functions: A dynamic dual pathway model," *Brain and Language*, vol. 89, pp. 267–276, 2004.
3. Korinna Eckstein and Angela D. Friederici, "Late interaction of syntactic and prosodic processes in sentence comprehension as revealed by erps," *Cognitive Brain Research*, vol. 25, pp. 130–143, 2005.
4. M. R. Mayberry III and Risto Miikkulainen, "SARDSRN: a neural network shift-reduce parser," in *Proceedings of IJCAI-99*, pp. 820–825. Kaufmann, 1999.
5. Teuvo Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, 1984.
6. S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. 11, pp. 674–693, July 1989.
7. S. Kadambe and G.F. Boudreaux-Bartels, "A comparison of a wavelet transform event detection pitch detector with classical pitch detectors," *Twenty-Fourth Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1073–1078, 1990.

Phonetic Sequence to Graphemes Conversion Based on DTW and One-Stage Algorithms

Rafael Teruszkin[1] and Fernando Gil Vianna Resende Jr.[1,2]

[1] Programa de Engenharia Elétrica, COPPE, UFRJ
[2] Departamento de Engenharia Eletrônica e de Computação, Escola Politécnica, UFRJ
{rafaelt, gil}@lps.ufrj.br

Abstract. This work proposes an algorithm for converting phonetic sequences into graphemes using DTW on the recognition of isolated words or closed sentences, and using One-Stage on a continuous speech recognition task. Most speech recognition systems resolve the task of recognition on a single stage without having an intermediate phonetic sequence result. The proposed solution is hybrid in the sense that it uses HMMs and Viterbi Decoding for recognizing a phonetic sequence (actually, triphones) and then DTW or One-Stage to generate the corresponding graphemes. Experimental results showed an average accuracy rate of 100% on the recognition of closed sentences, and average word recognition rate of 84% for the continuous speech recognition task.

1 Introduction

Most of the available speech recognition systems resolve the task of recognition on a single stage without having an intermediate phonetic sequence result (the concept named multi-pass is currently used for describing systems that work with intermediate word graphs and apply different language knowledge on each pass). The presented solution is hybrid in the sense that it uses HMMs (Hidden Markov Models) and Viterbi Decoding [1] for recognizing a phonetic sequence (actually, triphones) and then DTW (Dynamic Time Warping) [2] or One-Stage [3] to generate the corresponding graphemes. This method brings flexibility for having on one side a general task triphone recognizer and, on the other side, a specialized text-only converter.

The experiments made for testing the Phonetic Sequence to Graphemes (PS2G) converters used the results obtained from a primary recognition system based on HTK [1], as indicated in Fig. 1. This primary system was trained and tested with a database built from a set of phonetically balanced sentences [4] which also originated the system vocabulary. The vocabulary words were then automatically transcribed [5] and a System Dictionary was built to store the transcriptions. Using this dictionary, the PS2G converter is able to receive the phonetic sequences outputted by HTK and tries to recognize the most similar graphemes associated to them.

The next sections are organized as follows. In Sections 2 and 3, we present the principles for the converters based on DTW and One-Stage. In Section 4, the results obtained with these converters are presented and, Section 5 brings some conclusions.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 220 – 224, 2006.

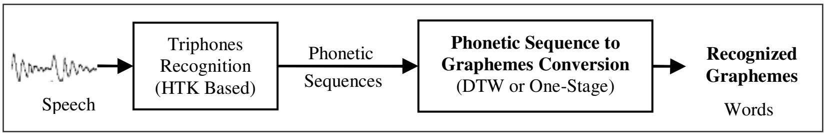

Speech
Triphones Recognition (HTK Based)
Phonetic Sequences
Phonetic Sequence to Graphemes Conversion (DTW or One-Stage)
Recognized Graphemes
Words

Fig. 1. Block diagram showing input and output of the PS2G converter

2 Conversion Based on DTW

The phonetic sequences received by the PS2G converter are the recognized triphones generated by HTK as a result of inputted speech samples of words or closed sentences that belongs to the vocabulary of the proposed system. Obviously, the outputted triphone sequences may contain errors as compared to the ideal transcription, and is the role of the converter recognizing the correct graphemes related to these sequences. To solve this problem, a variation of the DTW algorithm, used mainly for isolated word recognition tasks, was implemented. The idea of using DTW in those systems is its capacity of aligning different sounds on the time axis and then calculating a measure of distance (or cost) between them [2].

In the traditional DTW algorithm, the Itakura or Euclidian distance is calculated between coefficients vectors of frame *i*, from the test sound, and of frame *j*, from the reference sound. The modification made on the distance calculation of the proposed converter estimates the similarity between the phoneme *i* from the test word and phoneme *j* from the reference word, as seen in Table 1. That table shows that identical phonemes do not add any cost to the distance calculation. This relation can be latter modified in order do add intermediate values at different phonemes that sound similar.

Having the similarity calculation between phonemes resolved as shown in Table 1, the rest of the DTW algorithm follows on the same manner as if we were comparing two sequences of sounds (coefficients vectors). After all the distances are calculated between the test and all references phonemes, we can classify which of them is most similar to the test sound (the one that generated the phonetic sequence).

Table 1. Values used to calculate distance between test and reference phonemes (similarity)

Phoneme Comparison	**Resultant Distance**
Phonemes are equal	0
Phonemes are different	1

3 Conversion Based on One-Stage

In the continuous speech recognition task, the target for the PS2G converter is to detect the parts of the inputted phonetic sequence that are most similar to the references stored in the system dictionary. The set of references are then returned as the recognized graphemes. If no mistakes are made in this task, the graphemes are the words which had the samples inputted in the triphone recognition system. The solution for handling this conversion is an implementation of the One-Stage algorithm, which is one of the bases of modern continuous speech recognition systems.

The advantage of using the One-Stage algorithm is its ability to perform the three operations of word boundary detection, nonlinear time alignment and recognition simultaneously. In order to make the alignment between test and references sequences, K plans with dimensions I x J_k are created, where K is the number of references found in the system dictionary, I is the number of phonemes of the test sequence and J_k is the number of phonemes of reference with index k. During this alignment, two kinds of transition are allowed between the nodes (i, j, k). The first transition is very similar to the local restriction applied on DTW. It is allowed for nodes with j > 1 and represents the transitions from other nodes on the same plan which are on the left, bottom or diagonal relative positions. The second type of transition can happen to nodes with j=1 (reference borders). Those nodes can accept transitions from left nodes of the same plane or from nodes placed on superior left position of other plans, meaning a transition between references as stated in the next equation:

$$D(i,j,k) = \min \left\{ \begin{matrix} D(i-1,j,k) \ :: \text{left transition} \\ \min_{k_2}\left(D(i-1,J_{k_2},k_2) \right) \ :: \text{reference borders} \end{matrix} \right\} + d(i,j,k) \ . \tag{1}$$

The second minimization seen in (1) can also take into account the transition probability of reference k_2 to reference k, based on their occurrence on a training text. This probability is known as bigram and can be simply estimated in the following way [6]:

$$P(k \mid k_2) = \frac{1 + C(k_2,k)}{V + \sum_k C(k_2,k)} \ , \tag{2}$$

where $C(k_{2,} k)$ is the number of times that this transition is seen on a training text and V is the size of the system vocabulary.

After finishing the calculation of all nodes on each k plan, the algorithm finds the node with the smallest accumulated distance $D(I , J_k , k)$. From this point on, a backtrack procedure is executed until an initial point is reached [3]. All k references where the backtrack passes are recorded to generate the sequence of recognized graphemes.

4 Tests and Results

A database of 200 phonetically balanced sentences was used [4] for the experiments done in this work. Those sentences were recorded from a male speaker with a good microphone in a quite environment at 16 KHz, 16 bits per sample. From this set, 160 sentences were used for training the triphones of a continuous triphone recognizer based on HTK [1] using continuous HMMs and Viterbi Decoding. For this training, the sentences were manually segmented to phonetic sequences using the SAMPA Portuguese alphabet. The other 40 sentences were inputted to this recognizer and generated a set of recognized phonetic sequences. Those phonetic sequences were used for testing the PS2G converter within the DTW and One-Stage implementations.

4.1 Results of the Implementation Based on DTW

For the DTW implementation, the 40 phonetic sequences were tested against the ideal transcriptions generated for each sentence which formed a system vocabulary made of closed sentences. The result obtained was 100% of accuracy and can be explained by the big variability between the sentences which helps the DTW algorithm distinguishing them, even with the errors generated by the triphones recognition system.

4.2 Results of the Implementation Based on One-Stage

In the One-Stage implementation, the 181 different words which formed the 40 sentences were automatically transcribed and added to the system dictionary. As this is a continuous word recognition process, the accuracy is obtained with the words error rate (*WER*) measure like most of the systems do, and related word recognition rate [2].

As described before, the minimization used for calculating the cost of transition from one reference to the other could use or not the bigram probabilities as a transition penalty. Regarding that, the tests made for the converter based on One-Stage were done for both minimizations. The training text used to calculate the bigram probabilities was the same one of the testing sentences. For validating the implementation of the One-Stage algorithm, the converter was tested beforehand with the phonetic sequences generated automatically by the transcriber (ideal phonetic sequences). The results of these tests are summarized in Table 2.

Table 2. Summary of results (*WRR*) obtained with the PS2G converter based on One-Stage

Converter Input \ Minimization Mode	Minimization **without** bigrams	Minimization **with** bigrams
Ideal phonetic sequences	**96%**	**99%**
Output from the triphone recognizer	**50%**	**84%**

5 Conclusions

This work presented a PS2G conversion mechanism, based on DTW and One-Stage algorithms. The results obtained for the converter were 100% in the closed sentences recognition task using DTW and 84% in the continuous speech recognition task based on One-Stage using bigrams probabilities. Using bigrams made a huge difference on the converter performance increasing the accuracy in 68%.

Future work includes using a bigger database, training the bigrams with different texts than the ones used for testing the system and also enhancing the way that the phonemes are compared by the algorithms.

References

1. HTK - Hidden Markov Model Toolkit (http://htk.eng.cam.ac.uk).
2. Rabiner, L. R., Juang, B., *Fundamentals on Speech Recognition*, New Jersey, Prentice Hall, 1996.
3. H. Ney. “The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition”, *Proceedings of ICASSP*, 1984.

4. A. Alcaim, J. A. Solewicz e J. A. Moraes, “Freqüência de ocorrência dos fones e listas de frases foneticamente balanceadas no português falado no Rio de Janeiro”, *Revista da Sociedade Brasileira de Telecomunicações*, Rio de Janeiro, v. 7, n. 1, p. 23-41, 1992.
5. F. L. F. Barbosa, et al, “Grapheme-phone transcription algorithm for a Brazilian Portuguese TTS”, *Proceedings of PROPOR*, Faro, Portugal, 2003.
6. Huang X., Acero, A., Hon, H., *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall, ch. 11, 2001.

Very Strict Selectional Restrictions: A Comparison Between Portuguese and French*

Éric Laporte[1], Christian Leclère[1], and Maria Carmelita Dias[2]

[1] IGM, Université de Marne-la-Vallée - 5, bd. Descartes,
77454 Marne-la-Vallée CEDEX 2, France
eric.laporte@univ-mlv.fr, christian.leclere@univ-mlv.fr

[2] Pontifícia Universidade Católica do Rio de Janeiro,
Rua Marquês de S. Vicente, 225 - Rio de Janeiro - 22453-900, Brasil
mcdias@let.puc-rio.br

1 Introduction

We discuss the characteristics and behaviour of two parallel classes of verbs in two Romance languages, French and Portuguese. Examples of these verbs are Port. *abater [gado]* and Fr. *abattre [bétail]*, both meaning ‘slaughter [cattle]’. Such collocations are intermediate cases between verbal idioms and largely free verb phrases. Precise knowledge of these verbs would aid recognition of verb senses in texts and therefore be useful for natural language processing. The objective of this study is to compare the importance of these classes of verbs within the respective lexicon of both languages, and in particular to investigate corresponding pairs such as *abater [gado]/abattre [bétail]*.

2 Related Work

The distributional restrictions attached to word combinations show a continuum ranging from relatively free combinations between meaningful words (e.g. *mount a hill*) to the unique combinations observed between the elements of idioms (e.g. *mount the guard*). A distributionally frozen idiom such as *mount the guard* can be seen as an extreme case of selectional restriction in which only *the guard* can fill one of the slots of the verb (Guillet, 1986). In the intermediate case, the set of nouns that can fill a slot is reduced to a semantically homogeneous set of few nouns. For example, the verb *mount* in *mount a show* has a precise meaning observed only with *play, opera* and a few other nouns denoting spectacles. A list of phrases corresponding to this situation, called 32R3, had actually been produced for French (Boons *et al.*, 1976), and was extended later by the LADL[1] to a list of 1003 verbs[2].

When a verb selects a small set of nouns as its possible complement, this set can sometimes be characterized as the set of the hyponyms of a noun. For example, all the complements of Port. *abater [avião, helicóptero]* ‘bring down [plane, helicopter]’ are probably *aeronave* ‘aircraft’ and its hyponyms. Thus, our research is also indirectly

* This work has been partially supported by the CNRS.

[1] Laboratoire d'automatique documentaire et linguistique, Université Paris 7, 1968-2000.

[2] This list is available on the web at http://infolingu.univ-mlv.fr/english: follow Linguistic Data, Lexicon-Grammar, View.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 225 – 228, 2006.

connected with ontologies (Gruber, 1993) and the semantic classification of nouns (Gross, Clas, 1997).

3 Comparison Between French and Portuguese

We borrowed the definition of French class 32R3 from Leclère (2002) and transferred it to Brazilian Portuguese[3]. Verbs in this class share some important features: e.g. they take only one essential complement and do not admit a sentential object or subject. These criteria define the classes as highly residual ones, i.e. they select verbs with few other interesting properties[4].

We made a cross-lingual comparison between samples of the French and Portuguese classes. Our starting point was the list of French verbs (Boons *et al.*, 1976) beginning with *a*, in its updated version, and their translation into Portuguese. We then looked up all the verbs starting with *a* in an important Portuguese dictionary (Houaiss, 2001) and satisfying the definition, and translated the Portuguese verbs into French. This process produced a list of 74 Portuguese candidates and 75 French candidates to be included in the class. These lists include the entries in *a* that are members of the class in one of the languages, and their translation into the other, if they exist[5]. However, collocations and other word combinations are language-dependent phenomena: even if a member of the class in one language has a translation into the other, this translation is not necessarily a member of the class in the other language. For example, Fr. *aplanir [difficulté]* can be translated into Port. *amainar [dificuldade]*, but Port. *amainar* can be used with much more various complements than Fr. *aplanir* (with this abstract meaning). Other verbs are only translated by syntactically different constructions[6]. Thus, among a total of 74 pairs (French and Portuguese verbs of the sample and their translation), 47 % are pairs of members of this class in both languages (see table).

Table – Verbs with very strict selectional restrictions in French and Portuguese (excerpt)

FRENCH		PORTUGUESE	
VERB	COMPLEMENT	VERB	COMPLEMENT
Abattre	avion...	abater	avião...
abattre	bétail...	abater	gado...
abjurer	foi...	abjurar	fé...
abolir	loi...	abolir	lei...
abroger	loi...	ab-rogar	lei...
boucler	ceinture...	afivelar	cinto...

[3] When a verb has several senses, we considered each of them as a separate lexical entry, e.g. Port. *abater [avião]* 'bring down [plane]' was considered distinct from *abater [gado]* 'slaughter [cattle]'.

[4] However, the phenomenon of very strict selection also occurs with other verbs, e.g. intransitive or ditransitive verbs, and is worth a systematic study.

[5] Fr. *aplatir [balle]*, a technical term of rugby, has no translation in Brazilian Portuguese.

[6] Examples: without complement (Port. *abortar [feto]*/Fr. *avorter*), with a prepositional object (Fr. *amender [loi]*/Port. *fazer uma emenda em [lei]*), or with two verbs (Port. *adernar [embarcação]*/Fr. *faire gîter [embarcation]*).

We found that 62 % of the 74 Portuguese verbs, and 85 % of the 75 French verbs could be included into the class. This difference tends to indicate that though it is present in both languages, the phenomenon could affect more entries in French than in Brazilian Portuguese.

4 The Sets of Nouns Selected

The verbs in class 32R3 and in its Portuguese counterpart select small sets of nouns. These sets of nouns are by definition semantically homogeneous. For example, the complements of Port. *abater [avião]* 'bring down [plane]' are *aeronave*, *avião, helicóptero* and others. Some of these sets of nouns are even sets of synonyms: for example, Port. *amortecer [impacto]* takes such complements as *choque, colisão, impacto...* How can these sets of nouns be represented in a formal grammar? The conclusions of Guillet (1986) about French can be extended to Portuguese. Some of the sets of nouns in question can be specified through the choice of canonical representatives, e.g. *aeronave* for the complements of *abater [avião]*, i.e. these sets consist of the canonical representative and all its hyponyms. For example, the set of hyponyms of *aeronave* is the set of nouns *N* for which the sentence *N é um aeronave* '*N* is an aircraft' is true for the common sense[7]. However, in some cases, the only suitable canonical representative would be a whole phrase including elements of definition. For example, the comple-ments of Fr. *abattre [bétail]* 'slaughter [cattle]' denote domestic animals bred for certain purposes, but no French noun means this: *bétail* 'cattle', which excludes poultry, is two narrow, and *animal domestique* 'domestic animal', which includes pets, is too general. In other cases, it is difficult to find even a periphrasis that adequately defines the set of nouns. This is the case of Fr. *applaudir [spectacle, discours]* and Port. *aplaudir [espetáculo, discurso]* 'applause [show, speech]', though the meaning of the verb is the same for a show and for a speech.

5 Conclusion

The class of verbs we discussed are quite frequent in Portuguese and French, and probably also in other languages. Therefore, their study is useful for a number of computer applications both monolingual and bilingual. We showed that the behaviour of these combinations in Portuguese and in French are quite similar, but that the combinations themselves are mirrored by equivalent ones in roughly half the cases. The sets of nouns selected by this type of verbs are interesting subjects of study themselves, but their representation by canonical representative is bound to be only an approximation in a large number of cases.

[7] This set certainly includes *avião* and *helicóptero*, but relevant questions are: does the set of hyponyms include all the complements of *abater [avião]*? do all the complements of *abater [avião]* belong to the set of hyponyms? If the answer to both questions is yes, *aeronave* becomes a good choice to define this lexical entry of *abater*.

References

Boons, J.-P., Guillet, A., Leclère, Ch. 1976. La structure des phrases simples en français. 2. Classes de constructions transitives. Research report, LADL, University Paris 7, 86 + 58 p.

Gross, G., Clas, A. 1997. Synonymie, polysémie et classes d'objets. *Meta* XLII, 1, pp. 147-154.

Gruber, T. R. 1993. A translation approach to portable ontologies. *Knowledge Acquisition* 5(2), pp. 199-220.

Guillet, A. 1986. Représentation des distributions dans un lexique-grammaire, *Langue française* 69, Paris : Larousse, pp. 85-107.

Houaiss, A. 2001. *Dicionário Eletrônico da Língua Portuguesa*. Rio de Janeiro : Ed. Objetiva.

Leclère, Ch. 2002. Organization of the Lexicon-Grammar of French verbs, *Lingvisticæ Investigationes* 25:1, Amsterdam/Philadelphia: John Benjamins, pp. 29–48.

Towards a Formalization of Tense and Aspect for the Generation of Portuguese Sentences

Michel Gagnon[1], Eliana de Mattos Pinto Coelho[2], and Roger Antonio Finger[3]

[1] École Polytechnique de Montréal
michel.gagnon@polymtl.ca
[2] Université de Montréal
coelho@iro.umontreal.ca
[3] Centro Universitário Positivo
rogerf@positivo.com.br

1 Introduction

In this paper, we present an implemented generator of sentences in Portuguese that takes into account tense and aspect, especially the compound forms with the auxiliaries *estar* and *ser*. In Portuguese, the interpretation of the auxiliary *estar* is very similar to the English auxiliary *to be* when it is combined with a gerund to produce the progressive form. But it has many different interpretations when combined with a verb in the past participle form. In fact, the aspect system in Portuguese is very complex. It has been studied by many people in an interpretation perspective (see [5, 3]). But if we consider the generation perspective, we did not find any work giving some insight on how to achieve it. Our work is thus a fist step in this direction. After a short introduction of the notion of aspect we adopted, we provide an overall description of our grammar.

2 Aspectual Class and Coercion

The notion of aspectual class used in our implementation is similar to the one presented in [6], which is itself inspired of [7]. In short, the conceptualization of eventualities forms a tripartite structure called *nucleus*. A nucleus contains a *culmination*, which is a punctual event that makes the transition between two states: a *preparatory phase* and a *resulting state*. Lexically, verbs are classified according to which part of a nucleus they refer to. One class contains states of various kind. In our application, only lexical states are relevant. These are the states directly denoted by the verb, like *to be worried* and *to be ill*, and *consequent states* which are the result of an event. The events of our interest are the following ones[1]: activities (durative eventualities that do not have a natural culmination, like *to sleep, to run*), accomplishments (durative eventualities that become true only when a culmination is reached, like *to write a report, to build a house*, and achievements (punctual events that mark a transition between a preparatory phase and a resulting state, like *to arrive at home, to escape*).

[1] Moens and Steedman description also contains *point events*, not considered here because they are not themselves states and do not imply a resulting state.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 229–232, 2006.

An important characteristic of the formalism proposed in [6], and which has been formalized by [1], is the principle of coercion. This principle uses some operators to transform an eventuality of one type into an eventuality of another type. In our implementation, we use the following operators: CONSEQ, PREP and PROG. The first one, CONSEQ, returns the consequent state. Here it will be applied only to achievements and accomplishments. The PREP operator returns the preparatory phase (which is itself an activity) of an eventuality. If applied to an achievement, it corresponds to the state just before this achievement. If applied to an accomplishment, it corresponds to the state obtained by removing its culmination point. The PROG operator returns the progressive state of the eventuality to which it is applied. This operator will be applied only to activities, unless it is combined with the operator PREP. For example, let e be an eventuality denoted by *to write a report*. Since it is an accomplishment, we cannot apply the operator PROG to it. Applying the operator PREP, we get the activity PREP(e), which repesents the writing of the report without having it completed. By applying the operator PROG to this activity, we get the progressive state PROG(PREP(e)), which is denoted by *writing the report*.

In Portuguese, we need an additional operator, to take into account a sentence like *Paulo esteve escrevendo o relatrio. (At some time Yesterday, Paulo was writing the report)*. which requires a perfective interpretation of a progressive, which has no equivalent in english (we give a very approximate translation). As argued in [4], it denotes a progressive considered from an external perspective, instead of the internal perspective used in the usual progressive from. One consequence of this is that the progressive situation is temporally included in the temporal location adverbial, contrarily to the other form, which implies that the temporal location adverbial is temporally included in the situation. Consequently, we need another operator, PERF, which will transform a situation into a perfective situation. By default, we will consider that states are imperfective, whereas the other types of eventualities are perfective.

For example, there is no way to express a situation PREP(e) in Portuguese, which requires an imperfective point of view on a preparatory state. To express this state, we must turn it into a progressive state: PROG(PREP(e)). On the other hand, the consequent state CONSEQ(e) may be expressed by a passive form: *o relatrio estava escrito* (the report was written). Another example is a situation at present time, which cannot be perfective.

3 Generation System

With a simple semantic model based on four aspectual operators, and a lexicon where verbs are characterized by four aspectual categories, we developed a grammar which generative capacity is summarized in Figures 1 and 2. In the input, one of two voices (passive and active) must be selected and a combination of the operators are applied to the eventuality to be expressed. With this information, the grammar rules, together with the aspectual category found in the lexicon, determine the form of the generated sentence. We give, for many combinations of operators and both voices active and passive, an example of sentence that could be generated. In some cases, no sentence could be generated. This is as expected, since not all operator combinations are meaningful in Portuguese. Let us first consider the state class, at Figure 1. By definition, a state

State	
Semantic form	Example
active	
e	Paulo estava desconfiado
PERF(e)	Paulo desconfiou
	Paulo esteve desconfiado
PERF(PROG(e))	Paulo esteve desconfiando
PREP(e)	*does not exist*
CONSEQ(e)	*does not exist*
passive	*impossible*

Activity	
Semantic form	Example
active	
e	Paulo acompanhou Maria
PROG(e)	Paulo estava acompanhando Maria
PERF(PROG(e))	Paulo esteve acompanhando Maria
PREP(e)	*does not exist*
CONSEQ(e)	*does not exist*
passive	
e	Maria foi acompanhada por Paulo
PROG(e)	Maria estava sendo acompanhada por Paulo
	Maria estava acompanhada por Paulo
PERF(PROG(e))	Maria esteve sendo acompanhada por Paulo
	Maria esteve acompanhada por Paulo
PREP(e)	*does not exist*
CONSEQ(e)	*does not exist*

Fig. 1. Summary of operator combinations for states and activities

Achievement	
Semantic form	Example
active	
e	Paulo chegou
PROG(e)	*does not exist*
PREP(e)	*cannot be expressed*
PROG(PREP(e))	Paulo estava chegando
PERF(PROG(PREP(e)))	Paulo esteve chegando
CONSEQ(e)	Paulo estava caido
passive	
e	A televisão foi ligada
PROG(e)	*does not exist*
PREP(e)	*cannot be expressed*
PROG(PREP(e))	*cannot be expressed*
CONSEQ(e)	A televisão estava ligada
PERF(CONSEQ(e))	A televisão esteve ligada

Accomplishment	
Semantic form	Example
active	
e	Paulo escreveu o relatório
PROG(e)	*does not exist*
PREP(e)	*cannot be expressed*
PROG(PREP(e))	Paulo estava escrevendo o relatório
PERF(PROG(PREP(e)))	Paulo esteve escrevendo o relatório
CONSEQ(e)	*cannot be expressed*
passive	
e	O relatório foi escrito
PROG(e)	*does not exist*
PREP(e)	*cannot be expressed*
PROG(PREP(e))	O relatório estava sendo escrito
CONSEQ(e)	O relatório estava escrito
PERF(CONSEQ(e))	O relatório esteve escrito

Fig. 2. Summary of operator combinations for achievements and accomplishments

does not have a preparatory phase neither a consequent state. It is thus with no surprise that nothing can be obtained by using the operators PREP and CONSEQ. Note also that there are two generated form for the perfective form. This phenomenon is in agreement with what we found in the corpus. More study is required to further distinguish these two forms. Finally, since a state does not imply an entity suffering some action, it cannot be used in a passive form. Activities share with states the fact that they do not have a preparatory or consequent phase. But, on the other hand, they accept the passive form. An interesting fact concerning the passive form is that two forms exists for the progressive. We found both in our corpus study, but most of the time, one is much more frequent thant the other one. Now considering achievements and accomplishments, we see that they are very similar. They do not accept a passive progressive form, but an accomplishment accepts it if applied to its preparatory phase. Another difference is that the consequent phase can only be expressed with achievements at the active voice.

4 Conclusion and Future Work

In a future work, our model must be tested with more linguistic data. A meticulous study of the aspectual nature of each Portuguese verb should be achieved. For this, there already exists outstanding works ([2] for example). An exhaustive study of verbs must be realized to validate it. Also, it would be interesting to see how it can be adapted to other languages like French and English. Our appproach presented is relatively simple and concerns a limited set of the Portuguese language, but has the merit of being entirely implemented.

References

1. P. Blackburn, C. Gardent, and M. de Rijke. Rich ontologies for tense and aspect, 1996.
2. F. da Silva Borba. *Dicionrio Gramatical de Verbos*. Editora Unesp, So Paulo, 1990.
3. D. M. de Sousa Marques Pinto dos Santos. *Tense and aspect in English and Portuguese: a contrastive semantical study*. PhD thesis, Instituto Superior Tcnico, Universidade Tcnica de Lisboa, 1996.
4. M. Gagnon, E. Godoy, and R. de Oliveira. An implementation of DRT for a compositional implementation of the progressive in portuguese. In *4th International Workshop on Computational Semantics*, Tilburg, 2001.
5. R. Ilari. *A expressão do tempo em português*. Editora Contexto, São Paulo, 1997.
6. M. Moens and M. Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2), 1988.
7. Z. Vendler. *Linguistics in Philosophy*, chapter Verbs and Times. Cornell University Press, Ithaca, 1967.

The Need for Application-Dependent WSD Strategies: A Case Study in MT

Lucia Specia[1], Gabriela Castelo Branco Ribeiro[2], Maria das GraçasVolpe Nunes[1], and Mark Stevenson[3]

[1] ICMC – University of São Paulo, Av. do Trabalhador São-Carlense, 400,
São Carlos, 13560-970, Brazil
{lspecia, gracan}@icmc.usp.br
[2] DL - Pontificial Catholic University – Rio, R. Marquês de São Vicente, 225 – Gávea,
Rio de Janeiro, RJ, Brazil, 22453-900
gabrielacastelo@globo.com
[3] DCS – University of Sheffield, Regent Court, 211 Portobello Street,
Sheffield, S1 4DP, UK
M.Stevenson@dcs.shef.ac.uk

Abstract. It is generally agreed that the ultimate goal of research into Word Sense Disambiguation (WSD) is to provide a technology which can benefit applications; however, most of the work in this area has focused on the development of application-independent models. Taking Machine Translation as the application, we argue that this strategy is not appropriate, since many aspects of algorithm design, such as the sense repository, depend on the application. We present evidence for this by investigating the disambiguation of nine verbs in English-Portuguese translations.

1 Introduction

It is generally agreed that Word Sense Disambiguation (WSD) is more useful when it is meant for a specific application [9] but is usually treated as a general task; focusing on monolingual (particularly English) contexts using generic sense repositories, often WordNet [7]. For example, when used for Machine Translation (MT), a common strategy is to employ a generic WSD system and map monolingual senses into the target translations.

We consider WSD to be an intermediate task, and thus many key factors are specific for particular applications. In this paper we focus on identifying an appropriate WSD sense inventory for MT. We contrast the WordNet inventory for English disambiguation with the Portuguese translations assigned to a set of nine ambiguous verbs. Our goal is to show that there is not a one-to-one relation between the number of senses and translations, and that it is not due to the level of refinement of WordNet. With that, we provide evidence that employing monolingual WSD methods for MT is not appropriate.

Differences between sense repositories for monolingual and multilingual WSD have been investigated by others, with the distinct goals of either using monolingual WSD methods to carry out multilingual WSD [6], or mapping senses from one lan-

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 233 – 237, 2006.

guage into senses of another language [1]. Conversely, other approaches have employed multilingual information in the form of parallel corpora to support monolingual WSD [3, 4, 5, 8].

2 Contrasting Senses and Translations

In our experiments we analyze nine frequent and ambiguous verbs: “to come”, “to get”, “to give”, “to go”, “to look”, “to make”, “to ask”, “to live”, and “to tell”. Sentences containing these verbs and their phrasal verbs were collected from SemCor, Senseval-2 and Senseval-3 corpora (http://www.cs.unt.edu/~rada/downloads.html), in which words are annotated with WordNet 2.0 senses. In order to make the experiment feasible, a human translator filtered these sentences to extract 10 occurrences of each sense of each verb, and two occurrences of each sense of each phrasal verb. Excessively long and complex sentences and sentences with ambiguous uses of the verb were not selected.

Sentences were then distributed into five sets so that each set contained at least one occurrence of all senses of each verb, whenever possible, and a similar total number of sentences, varying from 289 to 309. Sense annotations were hidden and sentences were randomly ordered. In the resultant corpus, the number of senses for the nine verbs, along with their possible translations given by bilingual dictionaries, is shown in Table 1.

The selected sentences were then given to two groups of five professional translators, all native speakers of Portuguese. Both groups received the same sentences, so that we could investigate the agreement among translators. The translators were provided with entire sentences but were asked to translate only the verb.

In order to analyze the translations returned by each group, for each verb, we first grouped all the occurrences of an English sense and checked whether there were synonyms in the translations, which were considered a single translation. We then examined the relation between senses and translations, focusing on two cases: (1) if a sense had one or many translations; and (2) if a translation referred to one or many senses, i.e., whether one English sense was shared by many translations. Each sense was placed into two of the following categories: (a) or (b), mutually exclusive, representing case (1); and (c), or (d), also mutually exclusive, representing case (2):

(a) 1 sense → 1 translation: all the occurrences of a sense being translated as the same Portuguese word; e.g.: “ask” (“inquire, enquire”) is always translated as “perguntar”.
(b) 1 sense → *n* translations: occurrences of a sense being translated as different words; e.g.: “look” (“perceive with attention”) can be translated as “olhar” and “assistir”.

(c) *n* senses → 1 translation: different senses of a word being translated as the same Portuguese word; e.g.: “take advantage” (“draw advantages from” and “make excessive use of”) is translated as “aproveitar-se”.
(d) *n* senses → *n* translations: different senses of a word being translated as different Portuguese words; e.g.: two senses of the verb “run” (“move fast” and “carry out a process or program”) are translated as “correr” and “executar”.

Items (a) and (d) represent cases where multilingual ambiguity only reflects the monolingual one, i.e., to every sense of an English word corresponds a specific Portuguese translation. On the other hand, items (b) and (c) present evidence that multilingual ambiguity differs from monolingual ambiguity. Item (b) indicates that different criteria are needed for the disambiguation, as ambiguity arises only in the translation. Item (c) shows that disambiguation is not necessary, either because the translation is also ambiguous, or because Portuguese has a less refined sense distinction. We concentrate our analysis in cases (c) and (b).

Table 1. Verbs, senses in WordNet and possible translations in bilingual dictionaries

	ask	come	get	give	go	live	look	make	tell
# Senses	15	125	179	104	143	16	37	104	13
# Translations	16	226	242	128	197	15	63	239	28

3 Results

In Table 2 we present the number of sentences analyzed for each of the verbs, the English (E) senses and (non-synonyms) Portuguese (P) translations. The fifth and sixth columns represent the percentage of occurrences of the two categories outlined in Section 3, which we are focusing on, that is (b) and (c), with respect to the number of senses (# Senses). % (c) shows the percentage of senses which share translations with other senses, while % (b) shows the percentage of senses with more than one translation. In the other columns, (b) aver. shows the average number of P translations per E sense, while (c) aver indicates the average number of E senses per P translation.

Table 2. Results for the first group of translators

Verb	# Sentences	# Senses	# Translations	% (b)	(b) aver.	% (c)	(c) aver.
ask	83	8	3	0	0	87.5	3.5
come	202	68	42	38	3.1	73.2	6.3
get	226	90	61	30	2.6	61.1	3.4
give	241	57	12	51.3	3.3	84.2	6.3
go	231	84	54	39	2.9	76.2	4.4
live	55	10	7	16.7	3.0	70	2.7
look	82	26	18	36.8	2.4	84.6	2.7
make	225	53	42	48.6	2.9	77.4	4.1
tell	73	10	10	62.5	2.8	60	4.0

In general, the number of senses in the corpus is greater than the number of translations. On average, the level of ambiguity decreases from 45.1 (possible senses) to 27.7 (possible translations). % (c) shows that the great majority of senses share translations with other senses. For all verbs, on average, one translation covers more than two senses ((c) aver.). This shows that the disambiguation among many senses is not necessary and may result in an error, if the wrong sense is chosen. Consequently, these sense distinctions in WordNet are too fine grained for MT between this language pair.

In addition, (b) provides evidence that certain senses need more than one translation to express their meaning. In these experiments, this happened for all the verbs except "to ask" (the least ambiguous). The percentage % (b) of senses with more than one translation is impressive for all the other verbs. Thus, the lack of disambiguation of a word during translation because that word is not ambiguous in the source language can result in very serious errors, which shows the problems of employing monolingual methods for multilingual WSD. Therefore, this item shows that the use of sense inventories that are specific to English-Portuguese translation would be more appropriate.

Results shown in Table 2 refer to the first group of translators. In an attempt to quantify the agreement between the two groups, we computed the Kappa coefficient, as defined in [2], separately for cases (1) and (2) presented in Section 2. In the experiment with case (1), groups were considered to agree about a sense of a verb if they both judged that the translation of such verb was or was not shared with other senses. In the experiment with case (2), groups were considered to agree about a sense if they both judged that the sense had or had not more than one translation. The average Kappa coefficient obtained was 0.66 for case (1), and 0.65 for case (2). These levels of agreement are satisfactory, if we compare them to the coefficient suggested in [2] (0.67), which has been adopted as a cutoff in Computational Linguistics.

4 Conclusions and Future Work

Experiments contrasting monolingual English WSD with WSD in English-Portuguese translation showed that there is not a one-to-one correspondence between the English senses and the Portuguese translations. In most of the cases, many senses were translated using a single Portuguese word. In other cases, different, non-synonymous, words were necessary to translate occurrences of the same sense, showing that differences between monolingual and multilingual WSD are not only a matter of the highly refined sense distinction criterion adopted in WordNet. As a consequence, these results reinforce our argument that applying monolingual methods for multilingual WSD can either imply unnecessary work, or result in disambiguation errors, and thus specific strategies must be used to achieve effective results in MT. We plan to carry out further investigations of the differences between monolingual and multilingual WSD contrasting the English senses and their translations into other languages, analyzing also nouns.

References

1. Bentivogli, L., Forner, P., and Pianta, E.: Evaluating Cross-Language Annotation Transfer in the MultiSemCor Corpus. COLING 2004, Geneva (2004) 364-370.
2. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. Computational Linguistics, 22(2) (1996) 249-254.
3. Dagan, I. and Itai, A. Word Sense Disambiguation Using a Second Language Monolingual Corpus. Computational Linguistics, 20 (1994) 563-596.
4. Diab, M. and Resnik, P.: An Unsupervised Method for Word Sense Tagging using Parallel Corpora. 40th Anniversary Meeting of the ACL, Philadelphia (2002).

5. Ide, N. Parallel Translations as Sense Discriminators. SIGLEX99 Workshop: Standardizing Lexical Resources, Maryland (1999) 52-61.
6. Mihàltz, M.: Towards A Hybrid Approach To Word-Sense Disambiguation in Machine Translation. Workshop Modern Approaches in Translation Technologies - RANLP 2005, Borovets (2005).
7. Miller, G.A., Beckwith, R.T., Fellbaum, C.D., Gross, D., and Miller, K.: Wordnet: An On-line Lexical Database. International Journal of Lexicography, 3(4) (1990) 235-244.
8. Ng, H.T., Wang, B., and Chan, Y.S.: Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. 41st Annual Meeting of the ACL, Sapporo (2003) 455-462.
9. Wilks, Y. and Stevenson, M.: The Grammar of Sense: Using Part-of-speech Tags as a First Step in Semantic Disambiguation. Natural Language Engineering, 4(1) (1998) 1-9.

A Lexical Database of Portuguese Multiword Expressions

Sandra Antunes, Maria Fernanda Bacelar do Nascimento, João Miguel Casteleiro,
Amália Mendes, Luísa Pereira, and Tiago Sá

Centro de Linguística da Universidade de Lisboa (CLUL),
Av. Prof. Gama Pinto, 2, 1649-003 Lisboa, Portugal
{sandra.antunes, fbacelar.nascimento, amalia.mendes,
luisa.alice, ptsa}@clul.ul.pt
miguel.casteleiro@zmail.pt

Abstract. This presentation focuses on an ongoing project which aims at the creation of a large lexical database of Portuguese multiword (MW) units, automatically extracted through the analysis of a balanced 50 million word corpus, statistically interpreted with lexical association measures and validated by hand. This database covers different types of MW units, like named entities, and lexical associations ranging from sets of favoured co-occurring forms to strongly lexicalized expressions. This new resource has a two-fold objective: to be an important research tool which supports the development of MW units typologies; to be of major help in developing and evaluating language processing tools able of dealing with MW expressions.

1 Introduction

Firth (1955) described a collocation as the characterization of a word according to the words that typically co-occur with it, showing that the meaning of a word is closely related to the set of co-occurring words and that the lexicon does not consist mainly of simple lexical items but appears to be populated with numerous chunks, more or less predictable, though not fixed. However, these word associations are not immediately identified when one only relies on intuition-based studies. But the availability of large amount of textual data and the advance of computer technologies allowed the development of corpus-based approaches which enable the identification and analysis of complex patterns of word associations, proving that, in fact, natural languages follow regular patterns at syntagmatic level.

Aiming to contribute to the study of MW expressions, this presentation will address an ongoing project, Word Combinations in Portuguese Language (COMBINA-PT), developed in order to account for the most significant MW units in Portuguese, extracted from a 50 million word balanced written corpus and imported into a lexical database. Specific issues will be addressed, like the corpus constitution (section 2), the MW unit's extraction tool (section 3), the process of selection of those units (section 4) and further developments (section 5).

2 Constitution of the Corpus

The extraction of significant word associations requires a large corpus of real-occurring data. The corpus used for MW unit's extraction is a balanced 50,8M word

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 238 – 243, 2006.

written corpus extracted from the Reference Corpus of Contemporary Portuguese, a monitor corpus of 330 million words, constituted by sampling from several types of written and spoken text and comprising all the national and regional varieties of Portuguese (http://www.clul.ul.pt/english/sectores/projecto_crpc.html). In the near future, we plan to enlarge our results by extracting the MW units of a Portuguese spoken corpus of 1M words, previously compiled at CLUL. However, the data will be processed separately due to the strong discrepancy between the available amount of written and spoken corpus.

Since a particular word may co-occur with different lexical units according to the type of discourse in which they occur, the corpus balance is an important aspect to be considered. In this way, it is essential that the different types of discourse have a balanced dimension in order to properly describe every different patterns of co-occurrence of a lexical unit. According to these criteria, the corpus has 50.866.984 tokens and has the following constitution: **Newspapers** (30,000,000); **Books** (10,818,719) of which Fiction (6,237,551), Technical (3,827,551), Didactic (852,787); **Magazines and Journals** (7,500,000) of which Informative (5,709,061), Technical (1,790,939); **Miscellaneous** (1,851,828); **Leaflets** (104,889); **Supreme Court Verdicts** (313,962); Revised transcriptions of the **Parliament sessions** (277,586).

3 Extraction of MW Units

The first step consisted on the extraction, from the corpus, of all the groups of 2, 3, 4 and 5 tokens with a minimum frequency of 3 for groups of 3 to 5 tokens and 10 for 2-token groups. This task was performed using a software developed at CLUL. The groups automatically extracted are statistically analysed using a selected association measure and are afterwards sorted. The tool allows the user to select which measure to apply, and was first run with Mutual Information (MI), that calculates the frequency of each group in the corpus and crosses this frequency with the isolated frequency of each word of the group, also in the corpus (Church & Hanks 1990).

In order to reduce noise and only extract as much relevant MW units as possible, several cut-off options were implemented when running the extraction tool: (i) excluding combinations separated by punctuation; (ii) excluding two-word groups with initial or ending grammatical words using a stop list; (iii) excluding groups under the selected total minimum frequency. The final candidate list obtained comprises 1,751,377 MW units, still a considerable number.

The results of the application of the MW unit's extraction tool are presented in table 1, exemplified by the MW unit *espécies selvagens* 'wild species'.

In the results presented in table 1, the tool automatically extracts several types of information: (i) Distance: groups of 2 tokens can be contiguous or be separated by a maximum of 3 tokens, while groups of more than 2 tokens are contiguous (first number after the MW unit in bold); (ii) Number of elements of the group ("eg"); (iii) Frequency of the group at a specific distance ("og"); (iv) Lexical association measure (Mutual Information) ("ic"); (v) Total frequency of the group in all occurring distances ("fg"); (vi) Frequency of each element of the group ("fe"); (vii) Total number of words in the corpus ("N"); (viii) Concordances lines (in KWIC format) of the MW expression in the corpus, together with reference code.

Table 1. Example of the extraction of the MW unit *espécies selvagens* ‘wild species’

# 15 **espécies selvagens** 1 eg(2) og(15) ic(9.845638) fg(15) fe(2066 397) N(50310890)		
o topo da tabela de animais de	espécies selvagens	que morreram
odo selectivo, acaba por afectar	espécies selvagens.	Por isso, é
ameaça à conservação de algumas	espécies selvagens,	como o abutr
qualidade do habitat das nossas	espécies selvagens	seja também u
carretar "enormes ganhos para as	espécies selvagens”.	Dizem que a
tenas de milhar de exemplares de	espécies selvagens,	tanto cinegé
vido verdadeiras carnificinas de	espécies selvagens	protegidas em
presentados actualmente por duas	espécies selvagens	e por raças d
nternacional sobre o Comércio de	espécies selvagens	(CITES). Uma
existe uma grande quantidade de	espécies selvagens.	Na calma mad
passar (muito facilmente!) para	espécies selvagens	semelhantes.
Os transgenes que passam para as	espécies selvagens	não podem dep
mbientes, com inúmeros habitats,	espécies selvagens,	recursos nat
izadas Base de Dados relativas a	espécies selvagens	de Flora e Fa
um homem suspeito de tráfico de	espécies selvagens.	Para apreend

4 Selection of MW Units

In order to enable the representation of MW units and to offer a platform for user-friendly manual validation a lexical database was designed in Access format. The candidate list is loaded into the database together with the associated fields: statistical measure, frequency, distance, number of elements and concordance lines in KWIC format. The manual revision process consists of MW expressions validation as well as concordance lines validation since some contexts are wrongly associated with specific MW expressions. Doubtful cases of significant concordance lines can be solved by viewing a larger concordance context since the database is associated with the corpus through an index file. When concordance lines are eliminated, the total group frequency is automatically recounted in the Frequency field.

When selected as a significant MW unit, the group is attributed a numeric value. The first objective was to establish a correspondence between numeric values and a MW units typology that would be based on cohesion, compositionality (or not), substitutability, etc. However, first experience of MW units validation proved to be extremely difficult to establish the degree of fixedness of each group and a very time-consuming task to be performed at a first stage of the work. In order to accomplish this task in the established time and with the maximum accuracy, the decision was taken to only select the groups that presented some syntactic and semantic cohesion. This filtration will afterwards allow an easier elaboration of a more precise typology of MW expressions.

According to these criteria, when a MW unit is selected as a valid one it receives a value for the attribute “Type of multiword unit”, that covers the following types:

- groups forming a lexical category (e.g., *chapéu de chuva* 'umbrella'; *casa de banho* 'bathroom'; *fim de semana* 'weekend'; *fato de banho* 'swimming suit' – these are examples of expressions that may or may not occur with an hyphen);
- groups forming a phrase, for example a nominal or adjectival phrase, and presenting different degrees of fixedness (e.g., *senso comum* 'common sense'; *arte contemporânea* 'contemporary art'; *manteiga rançosa* 'rancid butter'; *onda/maré/vaga de assaltos* 'wave of robberies; *fogo/lume brando* 'gentle/soft fire'; *países/estados membros* 'member states');
- groups that constitute a verbal phrase or a sentence, with different degrees of cohesion (e.g., *olhar de lado* 'to leer at'; *lançar um ultimato* 'to make an ultimatum'; *pôr a mesa* 'to set the table'; *recuperar de uma lesão* 'to recover from a injury'; *um dia é da caça outro do caçador* 'every dog has his day');
- groups that specify named entities, such as institutions, functions, etc. (e.g., *União Europeia* 'European Union', *Presidente da República* 'President of the Republic'; *Dia Mundial da Paz* 'International/World Day of Peace; *Tratado de Amesterdão* 'Amsterdam treaty');
- cases that require further attention because they are doubtful MW expressions or because the MW unit has more than 5 tokens (maximum of tokens extracted from the corpus) and will be later recovered (e.g.; *não há amor como o primeiro* 'there's no love like the first love'; *pôr/colocar o carro à frente dos bois* 'to put the cart before the oxen').

The large candidate list of 1,7 million units made it impossible to assure manual validation of the whole data in the two-year time of the project, making it necessary to hand-check only a subpart of the groups automatically extracted. The selection of this subpart relied on the association measure applied to the set of candidate list, namely Mutual Information (MI), and on the results obtained with the list ordering. Our previous work on corpus extracted MW expressions (Bacelar do Nascimento (2000) and Pereira & Mendes (2002)) using MI had showed the strong tendency of this measure to present the best results with medium values instead of presenting the most significant MW units at the top of the results and new observation of specific lemma confirmed this evaluation of MI measure (similar to the conclusions attained by evaluative studies of several word association measures like Evert & Krenn (2001)). The total candidate list presented MI values between -5 and 33 and data observation showed that the most significant MW units received a MI value between 7 and 11. Having in mind the time-span available for manual validation, we selected a first subpart of the candidate list covering MI values between 8 and 10, in a total of 170,000 units, i.e., 10% of the initial candidate list, which would be hand-checked. From these 170,000 units, we selected about 31,000 as being significant MW units and other 1,637 groups were considered doubtful and will be further evaluated.

A list of all the word forms present in a selected MW unit is automatically created enabling the evaluation of all the groups a word enters in and producing a list of lexical elements associated with all the MW expressions that contain that word and that were considered significant units. Manual validation can also be processed through the list of all inflected forms in the candidate list, since each inflected form is associated with a list of all MW expression it enters in.

All the word forms that are part of the MW expressions that were manually validated as significant ones are compiled into a list of lemma and the hand-checking process will next proceed by covering all MW expressions of those lemma in order to achieve lexical association coverage for specific lexical elements.

5 Further Developments

The program will, in a latter stage, be run with other lexical association measures like t-test and log-likelihood (Dunning, 1993), using the set of already manually validated MW expressions as an important source of data for results evaluation and association measures comparison.

The syntactic and semantic analysis of the selected list of units will be the basis for proposing a typology of MW expressions that will build on the large set of real-occurring data from the corpus. Besides the issues on fixedness degree and compositional meaning, the study of these MW expressions will allow the identification of associative patterns like: (i) co-occurrence patterns (systematic co-occurrence with particular lexical items in a contiguous or non-contiguous form); (ii) grammatical patterns (systematic co-occurrence with a certain verb class, with a specific temporal verb forms or with certain syntactic constructions); (iii) paradigmatic patterns (hyperonymy, homonymy, synonymy or antonymy phenomena); (iv) discursive patterns (strong associations in one language register can be a weak association in another register).

This lexical database will be an important resource for the Portuguese language that will make available important data for the study of MW expressions, from the point of view of lexicography and lexicology, the lexicon-syntax interface and natural language processing.

The Lexical Database of manually revised MW units will be available for online query at the project site (http://www.clul.ul.pt/english/sectores/-projecto_combina.html).

Acknowledgments

The work described in this presentation has been undertaken under the project Word Combinations in Portuguese Language (COMBINA-PT), sponsored by the Portuguese Ministry of Science (POCTI/LIN/48465/2002) and developed at the Center of Linguistics of the University of Lisbon.

The authors would like to thank the reviewers for all the helpful comments.

References

Bacelar do Nascimento, M. F. (2000) "Exemples de combinatoires lexicales établis pour l'écrit et l'oral à Lisbonne", in Bilger, M. (ed.), *Corpus, Méthodologie et Applications Linguistiques*, Paris: H. Champion et Presses Universitaires de Perpignam 2000, pp. 237-261.

Bahns, J. (1993) "Lexical collocations: a contrastive view", *ELT Journal*, 47:1, pp. 56-63.

Calzolari, N., C. J. Fillmore, R. Grishman, N. Ide, A. Lenci, C. MacLeod & A. Zampolli (2002) "Towards Best Practice for Multiword Expressions in Computational Lexicons", *Proceedings of the Second International Conference on Language Resources and Evaluation*, Las Palmas, Canary Islands; Spain, 29 May – 31 May 2002, pp. 1934-1940.

Church, K. W. & P. Hanks (1990) "Word association norms, mutual information, and lexicography", *Computational Linguistics*, 16 (1), pp. 22-29.

Clear, J. (1993) "From Firth principles: Computational tools for the study of collocation", in Baker, M., G. Francis and E. Tognini-Bonelli (eds.) *Text and technology: In honour of John Sinclair*, Amsterdam, John Benjamins.

Dunning, T. (1993) "Accurate methods for the statistics of surprise and coincidence", *Computational Linguistics*, 19(1), pp. 61-74.

Evert, S. & B. Krenn (2001) "Methods for the Qualitative Evaluation of Lexical Association Measures", *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 188-195.

Firth, J. (1955) "Modes of meaning", *Papers in Linguistics 1934-1951*, London, Oxford University Press, pp. 190-215.

Heid, U. (1998) "Towards a corpus-based dictionary of German noun-verb collocations", *Euralex 98 Proceedings*, Université de Liège.

Kjellmer, G. A. (1994) *Dictionary of English Collocations*, Oxford, Oxford University Press.

Krenn, B. (2000a) "CDB - A Database of Lexical Collocations", *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece, 31 May – 2 June 2000, vol. II, pp. 1003-1008.

Krenn, B. (2000b) "Collocation Mining: Exploiting Corpora for Collocation Identification and Representation", *Proceedings of KONVENCS 2000*, Ilmenau, Deutschland.

Mackin, R. (1978) "On collocations: Words shall be known by the company they keep", in *Honour of A. S. Hornby*, Oxford, Oxford University Press, pp. 149-165.

Mel'cuk, I. (1984) *Dictionnaire explicatif et combinatoire du français contemporain*, Les Presses de L'Université de Montréal, Montréal, Canada.

Pearce, D. (2002) "A Comparative Evaluation of Collocation Extraction Techniques", *Proceedings of the Third International Conference on Language Resources and Evaluation*, Las Palmas, Spain, pp. 13-18.

Pereira, L. A. S. & A. Mendes (2002) "An Electronic Dictionary of Collocations for European Portuguese: Methodology, Results and Applications", in Braasch, A. & C. Povlsen (eds.), *Preceedings of the 10th EURALEX International Congress*, Copenhagen, Denmark, vol. II, pp. 841-849.

Sag, I., T. Baldwin, F. Bond, A. Copestake & D. Flickinger (2002), "Multiword Expressions: A Pain in the Neck for NLP", in Gelbukh, A. (ed.) *Proceedings of CICLing-2002*, Mexico City, Mexico.

Sinclair, J. (1991) *Corpus, Concordance, Collocation*, Oxford, Oxford University Press.

Dedicated Nominal Featurization of Portuguese

António Branco and João Ricardo Silva

University of Lisbon, Department of Informatics,
NLX—Natural Language Group
{ahb, jsilva}@di.fc.ul.pt

Abstract. A widespread assumption about the analysis of inflection features is that this task is to be performed by a tagger with an extended tagset. This typically leads to a POS precision drop due to the data-sparseness problem. In this paper we tackle this problem by addressing inflection tagging as a dedicated task, separated from that of POS tagging. More specifically, this paper describes and evaluates a rule-based approach to the tagging of Gender, Number and Degree inflection of open nominal morphosyntactic categories. This approach achieves a better F-measure than the typical approach of inflection analysis via stochastic state-of-the-art tagging.

1 Introduction

Inflective languages pose a problem for current stochastic taggers as besides the usual POS tags, tokens need also to be tagged with a variety of inflection information, such as the values for the features of Gender, Number, Degree, Case, Mood, Tense, etc. This requires an extended tagset [3] which usually leads to a lower tagging precision due to the data-sparseness problem.

In this paper, we address this problem by studying what can be gained when inflection tagging is envisaged as an autonomous task, separated from POS tagging. Nominal featurization is thus circumscribed as the task of assigning feature values for inflection (Gender and Number) and, if applicable, degree (Diminutive, Superlative and Comparative) to words from the nominal morphosyntactic open classes (Adjective and Common Noun).

In Section 2, the algorithm is outlined and evaluated. In Section 3, concluding remarks are presented.

2 Algorithm

The morphologic regularities found in Portuguese suggest a straightforward rule-based algorithm for the autonomous assignment of inflection feature values given that word terminations are typically associated with a default feature value. For example, most words ending in `-ção`, like `canção` (*Eng.: song*) are feminine singular. Any exceptions to this can then be easily found by searches in machine-readable dictionaries (MRD): The exceptions are words with the designed termination but with inflection features that do not match the default one. For instance, `coração` (*Eng.: heart*) is masculine singular.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 244–247, 2006.

Assigning inflection features can thus be done by simply searching for a suitable termination rule and assigning the corresponding default inflection tag if the input token is not one of the exceptions to that rule.

However, using rules and exceptions is not enough to ensure that every token receives an inflection tag. The main reason for this is the existence of "uniform" words, which are lexically ambiguous with respect to inflection feature values. For example, `ermita` (*Eng.: hermit*), depending on its specific occurrence, can be tagged as masculine or feminine. By using nothing more than rules and exceptions, the output produced by the rule-based featurizer would always be `ermita/?S` (singular, but with an underspecified value for Gender).

To handle these cases, an algorithm can be envisaged that builds upon the fact that there is Gender and Number agreement in Portuguese, including within NPs. All words from the closed classes that have inflection features (Demonstrative, Determiner, Quantifier, etc.) are collected together with their corresponding inflection tags. During inflection analysis of a text, the inflection tags assigned to words from these closed classes are "propagated" to the words from open classes (Adjective and Common Noun) that immediately follow them. These may, in turn, propagate the received tags to other words.

The example below illustrates how tag propagation disambiguates an occurrence of `ermita` based on the Definite Article that precedes it. The tag given to `ermita` can then be propagated to the adjective `humilde` (*Eng.: humble*), which is also a uniform word.

`o/MS` → `ermita/MS` → `humilde/MS`
Eng.: the [masculine] humble hermit

`a/FS` → `ermita/FS` → `humilde/FS`
Eng.: the [feminine] humble hermit

In order to make a sensible use of this idea, one just has to ensure that tag propagation occurs only within NP boundaries. For that effect, some patterns of tokens and POS tags are defined such that, when they are found, tag propagation is prevented from taking place.

For example, propagation may be prevented from crossing the conjunction `e` (*Eng.: and*) or punctuation symbols such as the comma. This allows to properly handle propagation over an enumeration of NPs and other similar structures:

$$\overbrace{\texttt{cão/MS branco/MS}}^{\texttt{NP}}\ ,\ \overbrace{\texttt{gatas/FP pretas/FP}}^{\texttt{NP}}\ \texttt{e}\ \overbrace{\texttt{peixe/MS azul/MS}}^{\texttt{NP}}$$

Eng.: white dog, black cats and blue fish

Note that by using this featurization algorithm it is still possible for a token to be tagged with an underspecified inflection tag. This happens not only due to some propagations being blocked (the blocking patterns have a "defensive" design, preventing some correct propagations to avoid tagging in error), but also due to the so-called bare NPs, which do not have a specifier preceding the head Noun, as in `Eu detesto ermitas/?P` (*Eng.: I hate hermits*). It also occurs in non-bare NPs, provided that the specifier is itself a uniform word, such as `cada/?S` (*Eng.: each*), which is lexically ambiguous with respect to its Gender.

The underspecified inflection tags that still remain after this propagation algorithm has been run cannot be accurately resolved at this stage. At this point, one can take the view that it is preferable to refrain from tagging than to tag incorrectly, and not attempt to resolve the underspecified tags without certainty. The resolution of these tokens can be left to the subsequent phase of syntactic processing which, taking advantage of NP-external agreement,[1] may resolve some of these cases (more in Section 3).

2.1 Implementation

The list of ca. 200 terminations and corresponding default inflection values was built from a reverse dictionary. The exceptions to these rules were gathered by resorting to a MRD and finding entries with each one of the 200 terminations but with inflectional features that differ from the default. This lead to a list of ca. $9,500$ exceptions, with an average of 47.5 exceptions for each inflection rule.

To implement the propagation procedure described above, a lexicon with ca. $1,000$ words from closed classes and respective inflection features was collected by searches in MRDs for entries with the relevant POS categories. Additionally, 9 patterns for blocking feature propagation were required.

The algorithm was implemented using Flex,[2] which provides an easy way for patterns (defined by regular expressions) in the input to trigger actions.

2.2 Evaluation

The rule-based featurizer does not necessarily assign a fully specified feature tag to every token. Following [4, pp. 268–269] for the measures of recall and precision,[3] it is important to note that, by virtue of the design of the algorithm, a precision score of 100% can in principle be reached provided that the list of exceptions to termination rules is exhaustive. However, in our experiment, some errors were found as the MRD used to collect exceptions is not large enough. These missing entries were, however, not added to the exceptions list, as this provides a way to replicate our results with the MRD that was used.

The rule-based featurizer was evaluated over a corpus with ca. $41,000$ tokens, where ca. $8,750$ where Adjectives and Common Nouns.

Firstly, the featurizer was evaluated over an accurately POS-tagged corpus. In this way, POS tagging mistakes will not negatively influence the outcome of the featurizer. The evaluation was then repeated but now over an automatically POS-tagged corpus.[4]

When running over a correctly POS-tagged corpus, the featurizer is highly precise (99.05%). However, most application cases also require POS tagging to

[1] Agreement holding between Subject and Verb, Subject and predicative complement in copular constructions with *be*-like verbs, etc.

[2] Flex—Fast lexical analyzer generator: `http://www.gnu.org/software/flex`.

[3] Recall is "the proportion of the target items that the system selected" and precision is "the proportion of selected items that the system got right".

[4] The POS tagger used was TnT [2], with 96.87% accuracy. [1]

Table 1. Evaluation

	Correct POS			Automatic POS		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Stochastic	n/a	n/a	n/a	92.90%	100.00%	96.32%
Featurizer	99.05%	95.09%	97.03%	93.23%	95.06%	94.14%
w/ agreement	98.85%	99.88%	99.36%	93.55%	99.85%	96.60%

be done automatically. As the propagation mechanism is sensitive to POS errors, precision drops (to 93.23%) when running over an automatically POS-tagged corpus. These results are shown in Table 1.

3 Concluding Remarks

The main weakness of this rule-based approach is its recall score (around 95.1%), caused by the featurizer abstaining from tagging some uniform words in the hope that subsequent syntactic processing will solve those cases. An examination of a sample of 113 such cases showed that 97 of them could be resolved syntactically, leaving only ca. 16% of the underspecified tags still unresolved. Extrapolation from this result indicates that using syntactic processing to handle underspecified tags could lead to a great increase in recall. More specifically, taking the case of the featurizer running over automatically POS-tagged text, if only 16% of the underspecified tags were left unresolved, precision would be 93.55% and recall would increase to 99.85%, for an F-measure of 96.60%.

When compared with a typical stochastic approach, this featurization strategy turns out to be a better solution with better scores. In order to develop a tagger to assign POS tags extended with inflection values, we trained a tagger with TnT, that implements a HMM approach with back off and suffix analysis, and offers top scoring results for Portuguese tagging [1]. The resulting tool for inflection analysis presents an F-measure of only 96.32%. The results obtained are summarized in Table 1.

References

1. António Branco and João Silva. 2004. *Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese.* In Proceedings of the 4th Language Resources and Evaluation Conference (LREC). 507–510.
2. Thorsten Brants. 2000. *TnT—A Statistical Part-of-Speech Tagger.* In Proceedings of the 6th Applied Natural Language Conference (ANLP). 224–231.
3. Jan Hajič and Barbora Hladká. 1997. *Probabilistic and Rule-based Tagger of an Inflective Language: A Comparison.* In Proceedings of the 5th ANLP. 111–118.
4. Christopher Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press.

Part-of-Speech Tagging of Portuguese Based on Variable Length Markov Chains

Fábio Natanael Kepler and Marcelo Finger

IME – Instituto de Matemática e Estatística,
USP – Universidade de São Paulo

Abstract. Tagging is the task of attributing to words in context in a text, their corresponding Part-of-Speech (PoS) class. In this work, we have employed Variable Length Markov Chains (VLMC) for tagging, in the hope of capturing long distance dependencies. We obtained one of the best PoS tagging of Portuguese, with a precision of 95.51%. More surprisingly, we did that with a total time of training and execution of less than 3 minutes for a corpus of almost 1 million words. However, long distance dependencies are not well captured by the VLMC tagger, and we investigate the reasons and limitations of the use of VLMCs. Future researches in statistical linguistics regarding long range dependencies should concentrate in other ways of solving this limitation.

1 Introduction

An empiric approach to Natural Language Processing (NLP) suggests that we can learn the complicated and extensive structure of language by searching common patterns in its use, identified via Probability Theory. This leads to probabilistic models of linguistic phenomena whose parameters are induced by the application of statistical and machine learning methods.

This work concentrates on *part-of-speech analysis*, that is, attributing to words in context their correspondent part-of-speech (PoS) tag. Several computational linguistic tasks benefit from such analysis, such as parsing, automatic translation, grammar correction and information extraction.

Many words have more the one possible PoS tag, which is disambiguated by the context in which these words occur. However, even with contextual information some words remain ambiguous, and even an specialist could have difficulty in classifying them correctly.

Several computational approaches use statistical methods to create relations between words, tags and contexts via some form of supervised learning from a manually tagged corpus.

This work presents a study about the treatment of short, average, and long distance dependencies in the Brazilian Portuguese through the use of Variable Length Markov Chains (VLMC) [1, 2, 3]. We obtained accuracy results very near the best ones reported for Portuguese [4, 5, 6], and also with training and execution times well bellow the existing ones in literature. With respect to long

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 248–251, 2006.

distance dependencies, the VLMC was not able to detect very large contexts, and we investigate why this happens. Future works should be able to treat contexts of variable size and form.

The article is organized as the following. Section 2 shows some results we obtained with the VLMC-based tagger, regarding accuracy and execution time. Conclusions about the results and the applicability of the method are discussed in Section 3.

2 Tests and Results

We implemented the VLMC tagger using `C++` and `STL` (*Standard Template Library*), and compiled it with `g++` version 3.3.4. The tests were made on a machine equipped with one Intel Pentium 4 processor of 3 GHz, and 1 GB of RAM.

The tagger was trained and tested with the *Tycho Brahe* corpus [7], which contains various texts from historical Portuguese manually tagged, in a total of $1,035,593$ words. These words were splitted into a training corpus containing $775,602$ words, and a testing corpus containing $259,991$ words.

We executed sets of tests varying the size of the training corpus, choosing $5\%, 10\%, \ldots, 95\%, 100\%$ of its sentences and executing 10 times with each one of these sizes (randomizing the sentences each time), but always using the testing corpus without modifications.

Full results are presented in [8]. Here we concentrate on accuracy and execution time measures.

Figure 1 shows the accuracy[1] results for the tagger. Each point around displayed indicates the accuracy of one test iteration, and the curve itself crosses the average accuracy of each set of iterations. The final accuracy, when using 100% of the training corpus, is 95.51%. We can see that the greater the number of words used for training the smaller the difference of result between iterations of the same testing set. However, though the results may converge, this also shows that the tagger is sensible not only to the amount of training words, but also to the choice of the sentences. Some testing iterations achieved results far better or far worse than the average, indicating the existence of sentences that improve the tagger's learning or that worsen it.

Figure 2 shows three curves of times taken by the tagger in relation to the number of words used to train it: the total execution time, the time for training, and the time for tagging [2]. The total execution time when using 100% of the training corpus is 157 seconds. The three curves seems to show linear behaviour, and in fact the curve of the average time of execution shows correlation equal to 0.9969.

[1] By accuracy we mean the proportion of words from the testing corpus to which the tagger assigns the correct tag.

[2] Note that the total execution time is greater than the sum of the times taken in training and tagging because it includes the time spent by other operations such as reading the corpus' files and calculating the accuracy.

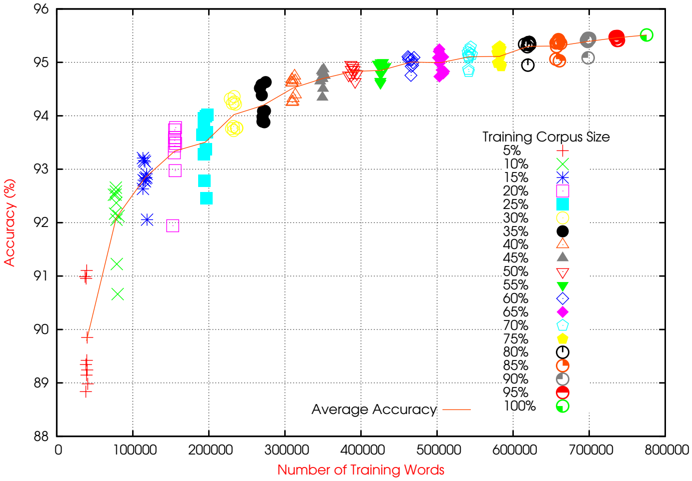


Fig. 1. Distribution of the tagger's accuracies

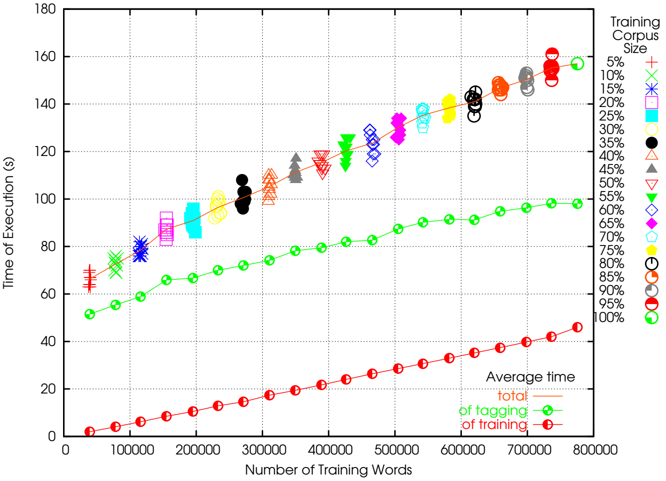


Fig. 2. Distribution of the tagger's execution times

3 Conclusions

Based on recent research about a theoretic statistical model called *Variable Length Markov Chains*, we have built a part-of-speech tagger for Portuguese.

With an accuracy of 95.51% we obtained a result very close to the best ones reported for Portuguese [4, 5, 6]. And with an execution time of 157 seconds using more than $775,000$ words for training and almost $260,000$ for testing ($1,035,593$ in total) we have obtained a time very inferior to the existing ones in literature[3].

When instructing the tagger to consider longer contexts, it was not able to detect many long distance dependencies. Moreover, instructing it to consider not so long contexts have improved the performance (in terms of accuracy). So we conclude that, when having less long contexts available, the tagger chooses short and recent contexts, what improves the performance and shows that there are long contexts that decay it.

We give results that allow us to observe limitations and advantages of the application of statistical models based on VLMC: they learn various short and average distance fixed contexts ($d \leq 6$), but they do not have generalizating capacity to learn linguistic phenomena occuring in variable contexts and of long distance. Future research in statistical linguistics regarding long range dependencies should concentrate in other ways of solving this limitation.

References

1. Schütze, H., Singer, Y.: Part-of-speech tagging using a variable memory markov model. In: Proceedings of ACL 32'nd. (1994)
2. Bühlmann, P., Wyner, A.J.: Variable length markov chains. Annals of Statistics **27**(2) (1999) 480–513
3. Mächler, M., Bühlmann, P.: Variable length markov chains: Methodology, computing and software. Research Report 104, Eidgenossische Technische Hochschule (ETH), CH-8091 Zürich, Switzerland (2002) Seminar fur Statistik.
4. Alves, C.D.C., Finger, M.: Etiquetagem do português clássico baseada em córpora. In: Proceedings of IV Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR99), Évora, Portugal (1999) 21–22
5. Aires, R.V.X.: Implementação, adaptação, combinação e avaliação de etiquetadores para o português do brasil. Dissertação de mestrado, Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo - Campus São Carlos (2000)
6. Finger, M.: Técnicas de otimização da precisão empregadas no etiquetador Tycho Brahe. In: Proceedings of V Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR2000), Atibaia, Brazil (2000) 19–22
7. IEL-UNICAMP and IME-USP: Corpus Anotado do Português Histórico Tycho Brahe. (2005) Acessado em 2005.
8. Kepler, F.N.: Um etiquetador morfo-sintático baseado em cadeias de markov de tamanho variável. Dissertação de mestrado, Programa de Pós-Graduação em Ciência da Computação, Instituto de Matemática e Estatística, Universidade de São Paulo (2005)

[3] Though time measures are relative to the machine used, the one we used has a domestic configuration.

From Syntactical Analysis to Textual Segmentation

Ana Luisa Leal[1], Paulo Quaresma[1], and Rove Chishman[2]

[1] Departamento de Informática, Universidade de Évora, Portugal
[2] UNISINOS – Universidade do Vale do Rio de Sinos, S. Leopoldo, Brasil
{analu, pq}@di.uevora.pt, rove@unisinos.br

Abstract. In this work a proposal for automatic textual segmentation is described. The proposal uses the output of an automatic syntactic analyzer – Parser *Palavras* – to create textual segmentation. Parse trees are used to infer text segments and a dependency tree of the identified segments. The main contribution of this work is the use of the syntactic structure as source for the automatic segmentation of texts, as well, as the use of inference rules for the textual organization.

1 Introduction

This work presents a proposal to make textual segmentation from syntactic structures. We use the information given by the parser *Palavras* [1] about syntactic structures, and we use a set of rules to obtain the segments and to infer their structure. This structure is called a DTS – dependency tree of segments. We believe that from a DTS, the macrostructure can be constructed. The objective of the author can be recognized through the analysis of the macrostructure offered by these trees.

From the output of *Palavras* it is possible to identify segments, *the concrete manifestation of propositions*, which are *conceptual structures*, compounding the textual structure. Moreover it is possible to obtain the relations that are subjacent to this microstructure. The results show that the proposed approach is able to perform the textual segmentation through the information obtained from the parser *Palavras*. Moreover, the result of the syntactic analysis also establishes the hierarchy of the information.

2 Text, Propositions and Segments – An Unique Relation

2.1 Text – The Conception

Our direct goal in this study is to perform the textual segmentation from the text automatic syntactic analysis, but we may also point out that our major goal is the recognition of the discursive structure in the macropropositions of texts. In this context, we are necessary engaged with the definition of what we

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 252–255, 2006.

understand as being text. In our work, we define that text is the superficial realization of discourse. So, we recognize discourse as a complex entity that contains propositions in which the author expresses his/her objectives. Text is a concrete realization of the discourse.

2.2 Propositions

The discourse is structured in terms of propositions and, in this sense, it can be recognized as a *big proposition* – macroproposition – of the complete sense, where it is possible to observe units with smaller size that are related in the morphologic, syntactic and semantic levels, to compose the discursive structure. In our work, the propositions are inferred from the output of the parser *Palavras*, which presents as result the text syntactically analyzed and segmented in structural terms. The syntactic information produced by the automatic analyzer is used to identify and classify what can be consider as a segment of the text, which represents propositions of discourse.

2.3 Segments

The segments and subsegments are recognized from the results of the automatic syntactic analysis. The segments represent the propositions of the discourse and through them are established the syntactic and semantic relations that are responsible for the sense of that discourse. We assume that the segments are the smaller units of significance, and these smaller units are what establish the microstructural relations.The propositions are the conceptual forms and the segments are the superficial realization forms of those propositions. However, it is acknowledged that the delimitation of these minimal units of significance represents a serious difficulty to the work that involve the textual segmentation, see Pardo and Nunes [4].

Considering the segmentation process, Carlson and Marcu [2] have also proposed rules that have as base the syntax; they can be applied to texts of different typologies. The rules are normally related to clause/sentences such as: main clause; subordinate clause with discourse cue; complements of attribution verbs; coordinate sentences; temporal clause. The authors call the attention to the relative clauses, appositive and parenthetical, because they must be treated as embedded segments. The difference between our proposal and the proposal of Carlson and Marcu [2] is that with the result of the syntactic analysis produced by parser *Palavras* it is possible to identify the segments of the text and to make a more correct segmentation. After concluding the segmentation, we use rules to relate the identified segments that compound the complete text.

The segments and subsegments play different roles. These segments are identified by Mann and Thompson [3] as nucleus and satellite. There is not a preestablished order to the manifestation of the segments in the text, but there are restrictions in the way the relations are established among them. Considering the segments snd subsegments presented in a text, we stand out the existence of relations among them. These relations have been presented initially by Mann and Thompson [3], classified as Rhetorical Relations.

3 The Parser *Palavras* – A Proposal to Textual Segmentation

3.1 Segmentation

Our proposal is based in the process of segmentation. This process is structured from the the automatic syntactic analysis and it gives us the segments of the text and as a consequence the propositions of discourse. The parser *Palavras* generates the constitutive blocks of text from which are produced the dependency trees of segments – DTS – and later the rules that determine what is the rhetorical role of the proposition in the discourse.

The segmentation is the stage that precedes the formalization of the dependency trees of segments and of the rules that define and delimit the segments, and it is based in the results of automatic syntactic analysis. From the automatic segmentation, we use rules to identify the segments and its correspondent propositions. The rules represent the possible combinations between the segments and its boundaries, as well as, the relations syntactic-semantics that result of these combinations.

4 Hierarchy, Rules and Relations of Text

The notion of textual hierarchy is important in our work because it is directly related to the identification of the textual macroproposition. In order to recognize and explain the order of the segments it is important to explain how the structural mechanisms of the text are articulated and how they represent the objective of author.

The hierarchy observed in the textual organization gives data to the construction of the dependency trees of segments – DTS – through which we can identify the segments and subsegments. This identification is important to determine which is the main segment and which are accessory/secondary to the thematic chain. In the process of systematization of the rules it is possible to develop a strategy that recognizes only the main segment which composes the subject of the text.

4.1 Rules and Dependency Tree Segments

The rules that we propose to create DTS are built from the result of the segmentation given by the parser *Palavras*. From the syntactic data it is possible to identify the segments of text, and their structure. The tree shows that it is possible to identify the principal segments and the secondary segments, main nodes and the secondary nodes of tree. The identification of the main nodes and the secondaries represent structurally the hierarchy in the organization and disposal of the segments and subsegments. This structural representation is relevant in our work, because we intend to identify automatically the proposition of the discourse.

4.2 Rhetorical Relations

Rhetorical relations are syntactic-semantic mechanisms that appear in the interior of the propositions represented by the segments. These relations are responsible to the presentation of the information of text. In this work, we do not present in detail all the problems around the rhetorical relations, because we are focusing in the textual segmentation problem from an automatic syntactic analysis view point. However, we believe that the rhetorical relations can also be inferred using the same DTS structure as the main source of information.

The identification of the rhetorical relations between the propositions may help in the process of selecting and excluding the propositions that are not necessary to identify the *big proposition* – macroproposition – of the text, i.e., the structure that synthesizes the objective of the author of a specific text.

5 Final Remarks and Future Work

The research that we propose is extensive and it is related with different knowledge areas, and different cognitive levels. We believe that it is possible to develop a robust system that is capable of articulating these areas and levels. Some of the stages proposed already were concluded and showed satisfactory results. These results give us a good support to the continuity of the study. As final claim, we strongly believe that it is possible to use the text parsing structure to obtain segments and its correspondent propositions, to generate DTS, to infer rhetorical structures, and to obtain the text macrostructure.

References

1. E. Bick. *The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework.* Aarhus University Press, 2000.
2. L. Carlson and D. Marcu. Discourse tagging reference manual. Technical Report 545, ISI Technical Report ISI-TR-545, 2001.
3. W. Mann and S. Thompson. Rhetorical structure theory: toward a functional theory of text organization. Technical report, Technical ReportISI/RS-87-190, 1987.
4. M. Pardo, T.A.S e Nunes. Análise de discurso: Teorias discursivas e aplicações em processamento de línguas naturais. Technical Report 196, Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, 2003.

Syntactical Annotation of COMPARA: Workflow and First Results

Susana Inácio and Diana Santos

Linguateca, SINTEF ICT

Abstract. In this paper we present the annotation of COMPARA, currently the largest parallel corpora which includes Portuguese. We describe the motivation, give a glimpse of the results so far, and the way the corpus is being annotated, as well as mention some studies based on it.

1 Introduction

COMPARA (www.linguateca.pt/COMPARA/) is a large parallel corpus based on a collection of Portuguese-English and English-Portuguese fiction texts and translations, which has been developed and post-edited (or revised) ever since 1999 [1]. COMPARA has been designed with a view to be an aid in language learning, translation training, contrastive and monolingual linguistic research and language engineering.

In this paper, we present for the first time the syntactical annotation of COMPARA and its intellectual revision (or post-edition), after its automatic annotation with PALAVRAS [2] of Eckhard Bick and a post-processing similar to the one used by the AC/DC project [3].

We suggest how this work can be used to measure [4] both PoS annotation entropy and/or perplexity of the Portuguese language, and the amount of work involved in automatic annotation and its intellectual revision. We also mention other kinds of studies or applications that could profit from this annotation.

2 Motivation

As of today, COMPARA offers a lot of functionalities that we believe are original and useful, namely (a) kinds of search (according to alignment type, for translator's notes, reordered units, foreign words and expressions, etc.); (b) kinds of output provided (concordances, several kinds of distribution, parallel snapshot, etc.); and (c) kinds of subcorpus selection (language variety, individual texts, dates). A full description of the DISPARA system is provided in [5].

However, one of the most sought after options – well known from both the BNC [6] for English and the AC/DC for Portuguese – was the possibility to make queries also based on part of speech, lemma, morphological and syntactical information.

After working since November 2004 in annotating COMPARA, and with a set of precise guidelines [7] in place, albeit still under development, we can now announce

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 256 – 259, 2006.

that (the majority) of the Portuguese side of COMPARA contains (revised) PoS, lemma, and morphological information, and that annotation of the English side, using the CLAWS tagger [8], is planned to start soon.

Let us present some examples of new search functionalities, to give some flavour of what is now possible: for forms ambiguous between grammatical categories, it is possible to (1) ask for their part of speech distribution, or (2) select (bilingual) concordances only of one grammatical interpretation. One can (3) get all forms of a given verb occurring in COMPARA by just selecting its lemma, as well as (4) obtain the distribution of forms or lemmas in a particular tense or in a particular syntactical or translational context. [9] presents contrastive examples where different syntactic realizations are relevant.

3 Kinds of Studies Allowed by Annotated COMPARA

Already in 1993 the first quantitative studies about PoS ambiguity in Portuguese were published by Medeiros et al. [10] and work in that direction has continued, under different projects, reported in [11], [12] and [13]. Actual data related to annotation of COMPARA can be found in [4].

There are several ways to define (part-of-speech, or morphological) ambiguity: in the lexicon, out of context (as was done in [10-12] using the knowledge embedded in morphological analysers), providing therefore a measure of the work required by a parser; or in running text (in a large enough corpus), where one only considers as ambiguous forms which happen to have more than one interpretation in the corpus [4]. Obviously, these two kinds of measures provide superior and inferior limits to the ambiguity in practice.

Another kind of studies that COMPARA now allows is quantitative studies of translation patterns [14], until now difficult and time consuming, since they required manual selection and annotation.

Finally, we believe COMPARA to be large enough to furnish evaluation material for several NLP tasks such as word or sentence alignment, word sense disambiguation and even machine translation.

4 Workflow and Comparison with Floresta Sintá(c)tica

In order to have the corpus return reliable information, it is necessary to check the output of automatic systems that attempt to do the complex job of assigning in context the right syntactical information to texts in natural language.

There are, however, many ways to perform such revision task, so it is interesting to document the way we are working, contrasting it with another project also concerned with human annotation of text in Portuguese, Floresta Sintáctica [15,16]. Basically, we can say that the annotation of Floresta has proceeded in a depth-first way, with every syntactic detail checked and eventually corrected starting from the first sentence in the corpus, while the annotation of COMPARA took a breadth-first approach, starting with PoS annotation and proceeding from the most frequent to the least frequent items.

These choices were of course motivated by the different intended user models of the two corpora: people interested in Portuguese syntax and/or quantitative studies or training of parsers for Floresta, while a much broader range of users for COMPARA, probably interested in (contrastive) lexical studies as well.

A list of all forms (or lemmata) was created per major part of speech and one proceeds by revising all contexts in which these words occur (starting from the top of the list, the most frequent first). This results also in a very different documentation activity: while for Floresta every piece of information present has to be documented, and note that *constructions which [a]re individually very rare [a]re collectively quite common* [17], in COMPARA we were instead concerned with other kinds of information such as guidelines about how to decide on a particular PoS in context, which, as far as we know, have never been published for Portuguese before. Grammars tend to describe phenomena with clearcut cases, while heuristic rules, such as the following, document how decisions were taken in a particular annotation task.

When one form can be both nominal and adjectival, choose noun:
- when it functions as a vocative: PPEQ2(741): E disse-me ele: «Que quer você, **amigo**?
- when it refers to a profession or activity: PBMA3(555): -- No tempo em que eu era **administrador**....
When one form can be both verbal and adjectival, choose adjectival:
- when senses are different: EBDL3T1(773): Mentiroso extraordinariamente convincente, o Boon: mesmo após anos de convívio **chegado** conseguia levar-nos, ...
- when it is modified by an adverb: EBDL1T1(1350): Ela adormeceu com um ar bastante **satisfeito**.

Also, the Floresta team has primarily dealt with syntactic vagueness or ambiguity (involving more than one token), while in COMPARA we have exclusively dealt with PoS vagueness or ambiguity [18].

Acknowledgement

This work was partially supported by grant POSI/PLP/43931/2001 from Fundação para a Ciência e Tecnologia (Portugal), co-financed by POSI.

References

1. Ana Frankenberg-Garcia & Diana Santos. "Introducing COMPARA, the Portuguese-English parallel translation corpus", in F. Zanettin, S. Bernardini and D. Stewart (eds.), *Corpora in Translation Education*, Manchester: St. Jerome Publishing, 2003, pp. 71-87.
2. Eckhard Bick. *The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press, 2000.
3. Santos, Diana & Eckhard Bick. "Providing Internet access to Portuguese corpora: the AC/DC project", in Gavriladou et al. (eds.), *Proceedings of the Second International Conference on Language Resources and Evaluation, LREC2000* (Athens, 31 May-2 June 2000), pp. 205-210.

4. Diana Santos & Susana Inácio. “Annotating COMPARA, a grammar-aware parallel corpus”, *Proceedings of LREC 2006*, Genoa, Italy, May 2006.
5. Diana Santos. "DISPARA, a system for distributing parallel corpora on the Web", in Elisabete Ranchhod & Nuno J. Mamede (eds.), *Advances in Natural Language Processing (Third International Conference, PorTAL 2002, Faro, Portugal, June 2002, Proceedings)*, LNAI 2389, Springer, 2002, pp.209-218.
6. Guy Aston & Lou Burnard. *The BNC Handbook: Exploring the British National Corpus with SARA*, Edinburgh University Press, 1996.
7. Susana Inácio & Diana Santos. "Documentação da anotação da parte portuguesa do COMPARA". In progress. First version: 9 December 2005. http://www.linguateca.pt/COMPARA/DocAnotacaoPortCOMPARA.pdf
8. Rayson, Paul & Roger Garside. “The CLAWS Web Tagger”. *ICAME Journal* **22**. HIT-centre - Norwegian Computing Centre for the Humanities, Bergen, pp. 121-123.
9. Diana Santos. "Breves explorações num mar de língua". *Ilha do Desterro* (2006).
10. José Carlos Medeiros, Rui Marques & Diana Santos. "Português Quantitativo"*, Actas do 1.º Encontro de Processamento de Língua Portuguesa (Escrita e Falada) - EPLP'93* (Lisboa, 25-26 February 1993), pp. 33-8.
11. Diana Santos. "Português Computacional", in Inês Duarte & Isabel Leiria (orgs.), *Actas do Congresso Internacional sobre o português, 1994, Volume III*, Lisboa: Edições Colibri / APL, Junho de 1996, pp. 167-84.
12. Diana Santos, Luís Costa & Paulo Rocha. "Cooperatively evaluating Portuguese morphology", in Mamede et al. (eds.), *Computational Processing of the Portuguese Language, 6th International Workshop, PROPOR 2003,* Springer, 2003, pp. 259-66.
13. Luís Costa, Paulo Rocha & Diana Santos. "Organização e resultados morfolímpicos". In Diana Santos (ed.), *Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa*. No prelo.
14. Santos, Diana. *Translation-based corpus studies: Contrasting English and Portuguese tense and aspect systems*. Amsterdam/New York, NY: Rodopi, 2004.
15. Afonso, Susana, Eckhard Bick, Renato Haber & Diana Santos. ""Floresta sintá(c)tica": a treebank for Portuguese", in M.G. Rodríguez & C.P.S. Araujo (eds.), *Proceedings of LREC 2002,* (Las Palmas 29-31 May 2002), ELRA, 2002, pp.1698-1703.
16. Afonso, Susana. Árvores deitadas: Descrição do formato e das opções de análise na Floresta Sintáctica. http://www.linguateca.pt/Floresta/ArvoresDeitadas.pdf. In progress. First version, 2004.
17. Sampson, Geoffrey. “The role of taxonomy in language engineering”, *Philosophical Transactions of the Royal Society (Mathematical, Physical and Engineering Sciences)* **358**, 4, 2000, pp. 1339–5.
18. Santos, Diana. "The importance of vagueness in translation: Examples from English to Portuguese", *Romansk Forum* **5** (1997), Junho 1997, pp. 43-69.

A Complex Evaluation Architecture for HAREM

Nuno Seco[1], Diana Santos[2], Nuno Cardoso[3], and Rui Vilela[4]

Linguateca nodes of Coimbra[1], Oslo[2], Lisbon[3], and Braga[4]
nseco@dei.uc.pt, diana.santos@sintef.no, ncardoso@xldb.di.fc.ul.pt, ruivilela@di.uminho.pt

Abstract. In this paper we briefly describe the evaluation architecture and the measures employed in HAREM, the first evaluation contest for named entity recognition in Portuguese. All programs are publically available for experimentation.

1 Introduction

Named Entity Recognition (NER) is nowadays regarded as a fundamental building block in the larger endeavour of understanding natural language by the NLP comunity. The recognition of NER as a separate task started with the MUC conferences [1] (more precisely MUC-6) and has ever after been considered in other contests of the same kind (e.g. ACE).

HAREM features several original traits and provided the first state of the art for the field in Portuguese [2], joining 10 different NER systems for Portuguese. We took into consideration most of the points mentioned in [3] that should be considered in future evaluations, such as (1) *domain independence* of the systems being tested, (2) *portability*, meaning that systems could be fine-tuned (or retargeted) to specific class of events and (3) encouraging work on *deeper semantic understanding*. Our goal with HAREM was to take the first step towards building an evaluation framework that could facilitate all these aspects.

Participants had to tag a large and varied collection, with 1202 documents (over 466,000 words) from 8 different genres and several varieties of Portuguese, of which a smaller part (the HAREM Golden Collection, the GC henceforth) had been manually hand-coded by the organizers, according to detailed guidelines discussed with the participants. We conceptually separated the NER process in two phases (even if most systems do not implement it this way): one first identifies an NE and then attributes some meaning to it in accordance with the surrounding context. In HAREM we took the classification process a step further by including a *morphological* task where NEs were assigned their respective gender and number in context. Another feature worth noting is that semantic classification was divided in two conceptual steps (categories and types), in order to more precisely pin down the intended meaning of the NE (see [4] for details).

2 Evaluation Facets and Options

Evaluation was carefully studied in order to provide the participants with the most relevant information possible. We hold the view that ceiling effects are as

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 260–263, 2006.

relevant in evaluation contests as are baselines. Therefore, we were extremely careful in maintaining vagueness during manual annotation of the golden collecion, either by allowing an entity to be assigned several semantic categories, or by supporting alternative delimitations by employing an ALT tag (see [4] and [2] for details). Another interesting note is that we also allowed participants to choose a set of categories and types that they wanted to be evaluated in (the *selective* scenario).

All these options had significant consequences on the complexity of the resulting architecture, as can be seen in Figure 1. In a nutshell, we had to implement

- alignment of the system output with the golden collection, which includes finding the best alignment among all alternative choices of ALTs;
- restriction of comparison to different sets of categories (a kind of ontology mapping);
- several different evaluation measures to reflect all these subtle distinctions.

The outcome is a modular architecture for NER evaluation, providing a valuable resource for further studies in the field.

Evaluation was divided into three tasks each capturing different aspects of the NER problem, namely (i) Identification; (ii) Semantic classification; and (iii) Morphological classification. These and other aspects of evaluation and the correponding metrics, are clearly explained in [2]. It will suffice to say here that NEs in terms of **identification** can either be considered *Correct*, *Partially correct by Excess*, *Partially correct by Shortage*, *Spurious* or *Missing*. We note that partially correct NEs correspond to NEs that systems identified and that overlapped in terms of tokens with NEs in the GC. As will be mentioned in section 3, partial alignments can be evaluated in several ways.

Semantic evaluation in our framework was captured through the use of four different measures. Since in the GC we marked NEs with a category and a type, obviously two of those measures individually took into account each of these axes, in which the possible values are *Correct*, *Spurious* or *Missing*. The other two measures combined category and type in an overall score. In terms of **morphological** classification, NEs were scored as either *Correct*, *Partially correct*, *Wrong*, *Missing*, *Spurious* or *Overspecified*.

3 Software

Automation of the evaluation task was achieved through the use of specialized evaluation software. Since our evaluation scheme differs from the MUC scenario and consequently from their software [5], we had to implement our own evaluator. The software is implemented in a pipelined architecture, where all programs read a file (the initial or an already partially processed submission), perform some processing and output a new file, which could then be studied and searched for unexpected situations, initially unforeseen. The HAREM evaluation architecture is presented in figure 1.

The first step is to extract, from the HAREM collection, the subpart for which there is a key (the golden collection). After alignment we individually

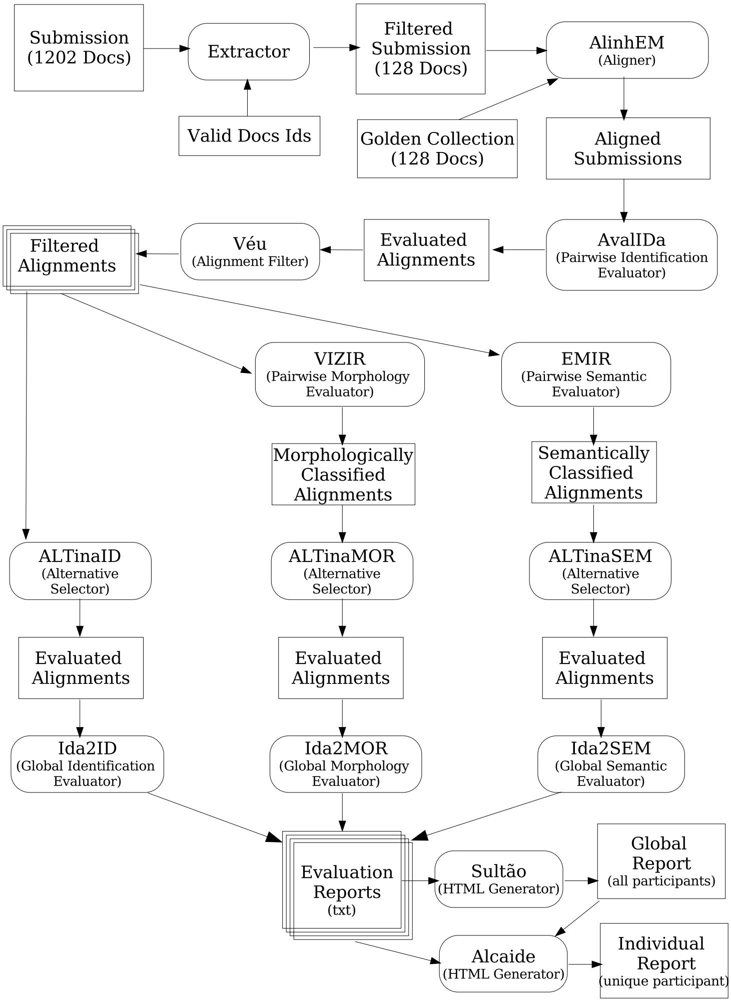

Submission (1202 Docs)
Extractor
Valid Docs Ids
Filtered Submission (128 Docs)
AlinhEM (Aligner)
Golden Collection (128 Docs)
Aligned Submissions
AvalIDa (Pairwise Identification Evaluator)
Evaluated Alignments
Véu (Alignment Filter)
Filtered Alignments
VIZIR (Pairwise Morphology Evaluator)
EMIR (Pairwise Semantic Evaluator)
Morphologically Classified Alignments
Semantically Classified Alignments
ALTinaID (Alternative Selector)
ALTinaMOR (Alternative Selector)
ALTinaSEM (Alternative Selector)
Evaluated Alignments
Evaluated Alignments
Evaluated Alignments
Ida2ID (Global Identification Evaluator)
Ida2MOR (Global Morphology Evaluator)
Ida2SEM (Global Semantic Evaluator)
Evaluation Reports (txt)
Sultão (HTML Generator)
Global Report (all participants)
Alcaide (HTML Generator)
Individual Report (unique participant)

Fig. 1. Overall HAREM Evaluation Architecture

evaluate each target NE the according to the rules described. In this step we simply attach information to every alignment attributing a score to each aligned NE.

Since we allowed participants to choose which NE categories they wanted to be evaluated in (the *selective* scenario), the alignment filter removes all the categories that should be ignored. Note that this step is not as trivial as may first seem, as we must account for situations where an NEs can be considered spurious or missing, depending on the type of filter. It is worthwhile noting that the filter can also be configured to ignore partial alignments (MUC style) or, in the case of several NEs aligning with one NE, just considering one NE as partially correct instead of all of them (we are grateful to Beth Sundheim (p.c.) for this suggestion). By default, our filter considers all partial alignments for evaluation purposes (for the exact metrics see `http://www.linguateca.pt/HAREM/`).

At this stage we have several files: one corresponding to the total scenario and the others to several selective scenarios (note that we also use the filter to select subparts of the task – by textual genre, by language variety, by single semantic category, etc). Each file produced will then follow three different evaluation paths. Since the alignments produced have only been individually evaluated according to the identification criteria, they have to be submitted to the pairwise morphology evaluator and the pairwise semantic evaluator as well.

Finally, after the best alternatives have been chosen, overall scores for precision, recall, over-generation and under-generation are computed for each task, and individual and global (comparative) HTML and PDF reports produced.

4 Concluding Remarks

Compared to MUC and other evaluation contests for NER, the architecture devised and deployed in HAREM represents progress, because it adds a number of degrees of freedom to experiment with. Some studies concerning Portuguese and different text genres have already been carried out.

More significant perhaps, is the fact that the whole architecture (and the programs implementing it) is publicly available. Together with the GC, any researcher can develop and test NER systems for Portuguese. Ample opportunity for reuse will, anyway, occur in the next HAREM contests.

Acknowledgements

This work was supported by grant POSI/PLP/43931/2001 from Fundação para a Ciência e Tecnologia, co-financed by POSI.

References

1. Hirschman, L.: The evolution of evaluation: lessons from the message understanding conference. Computer Speech and Language **12**(4) (1998) 281–305
2. Santos, D., Seco, N., Cardoso, N., Vilela, R.: Harem: An advanced NER evaluation contest for portuguese. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006, Genova, Italy (2006)
3. Grishman, R., Sundheim, B.: Message understanding conference-6: a brief history. In: Proceedings of the 16th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1996) 466–471
4. Santos, D., Cardoso, N.: A golden resource for named entity recognition for portuguese. In: This Volume. (2006)
5. Douthat, A.: The message understanding conference scoring software users manual. In: Proceedings of the 7th Message Understanding Conference (MUC-7). (1998)

What Kinds of Geographical Information Are There in the Portuguese Web?

Marcirio Silveira Chaves[1] and Diana Santos[2]

[1] Node of XLDB at University of Lisbon, Linguateca
[2] Node of Oslo at SINTEF ICT, Linguateca

Abstract. This paper presents some results about the geographical information in the Portuguese web and the overlap with people's and organization's named entities, using a geographic ontology based on authoritative sources and a named entity recognizer.

1 Introduction

This paper describes ongoing work on the study of a snapshot of the Portuguese web to assess how geographical information is encoded in the web in Portuguese (note that we are dealing here with a different web community and in a different language). We started by identifying and classifying named entities (NEs) focusing on locations.

Although place names are the kind of named entities which are easier to identify with the help of gazetteers (at least for English newspaper text [1]), the lack of a good named entity recognition (NER) system may hamper attempts to (naively) assign geographic scopes to web documents. For example, Zong et al. found that the ambiguity between geographic and non-geographic names was the main cause of errors [2].

2 Context and Available Resources

In order to conduct our study, we used several freely available resources, which we briefly describe in turn: in the GREASE project [3], associated to the tumba! search engine (`tumba.pt`), the first author and colleagues have created the Geographic Knowledge Base (GKB) [4]. GKB integrates items from multiple administrative information sources plus Internet data such as names of sites and domains in Portugal. Information in GKB about Portugal is exported as an ontology named Geo-Net-PT01 (`xldb.fc.ul.pt/geonetpt/`). Geo-Net-PT01 contains ca. 400,000 geographic terms and ca. 400,000 geographic relationships.

In addition, we used a Portuguese web collection called WPT 03 (`linguateca.di.fc.ul.pt/WPT03/`), with 12 Gbytes and 3.7 million pages comprising 1.6 billion words. Roughly 68.6% of these pages are in Portuguese.

In our experiments, we used the SIEMÊS NER system [5]. In the evaluation contest of NER for Portuguese, HAREM [6], SIEMÊS scores for the location category were close to 70% of precision and 75% recall. However, the version of SIEMÊS used in the our experiments is different from that used in HAREM.

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 264–267, 2006.

3 Getting to Know Better the Geographic References

The result of tagging the first randomly selected 32,000 documents with SIEMÊS is summed up in Table 1. We chose the three following categories of NE's: People, Organizations and Locations. SIEMÊS was configured to assign the People and Organizations because they are frequently ambiguous with or related to location terms, respectively. In Portugal, there are several surnames identical to location names, as in "Irene *Lisboa*" or "Camilo *Castelo Branco*." Also, we wanted to investigate how often a location was included in the name of an organization, to estimate how many cases it does provide a reliable clue to the physical place the organization is located in.[1]

Table 1. NEs detected in a 32,000 documents sample of WPT 03. MW stands for multi-word and GN stands for Geo-Net-PT01. DNE: Distinct named entities (types).

	# of NEs (%)	# of DNEs	# of MW NEs (%)	# of MW DNEs (%)	# of MW DNEs containing a name in GN (%)	# of DNEs occurring in GN (%)
PEO	250,585 (26.48)	77,228	140,155 (55.93)	58,991 (76.39)	24,105 (31.21)	521 (0.67)
ORG	418,915 (44.27)	114,353	214,698 (51.25)	89,790 (78.52)	26,789 (23.43)	462 (0.40)
LOC	276,775 (29.25)	47,972	90,018 (32.52)	36,395 (75,87)	22,959 (47.86)	4,576 (9.53)
Sum	946,275 (100.00)	239,553	444,871 (47.01)	185,176(77.30)	73,853 (30.83)	5,559 (2.32)

Table 1 shows that close to 1 million of NEs, belonging to the three categories, were identified, 30% of which corresponding to locations. For all categories, more than 75% of DNEs are multi-word. Organization names were the most frequent as far as tokens and types are concerned. As to type/token ratio, people NEs were the most varied, while locations displayed the lowest variation (i.e., location names were considerably more repeated in the sample than person names).

The last two columns of Table 1 measure partial and total overlap: while ambiguity with people's or organization's names is less than 1%, as much as 31.21% of the person DNEs and 23.43% of the organization DNEs contain a geographic name included in Geo-Net-PT-01.[2]

As to overlap of geographic locations in the web and in Geo-Net-PT-01, the numbers are astonishing at first: considering that Geo-Net-PT-01 is supposed to be complete as to Portuguese administrative geography, why only around 10% of the distinct locations present in the Portuguese web should appear in Geo-Net-PT-01? Even taking into account spelling errors or non-official naming conventions it is hard to account for the other 90%.

[1] For example, *Universidade do Porto* entails that it is located in *Porto*, while *Associação de Amizade Portugal-Itália* has no relation with location (the name refers to friendship among the peoples of the two countries) and this is even less so in the case of *Pastelaria Finlândia*, a name for a pastry shop.

[2] For this comparison, we used all names in Geo-Net-PT-01 (27,855), except for street names and postal codes.

Table 2. Distribution of the types contained in the local (LOC) category

Type	# of DNEs(%)	# of MW DNEs(%)
POV (names of pop. places)	33,827 (70.51)	24,037 (71.06)
ENDRALAR (full address)	3,505 (7.31)	3,313 (94.52)
SOCCUL (society/culture)	3,474 (7.24)	3,161 (90.99)
PAIS (country)	1,987 (4.14)	1,419 (71.41)
RLG (religion)	1,197 (2.50)	1,113 (92.98)
Other ($\sum$ 11 types)	3,982 (8.30)	3,352 (84.18)
Sum	47,972 (100,00)	36,395 (75,87)

Table 3. Distribution of NEs per document

	Total	Distinct		Total	Distinct
Avg. PEOs. per doc. with PEOs.	11.65	7.82	Median LOCs	4	3
Avg. ORGs. per doc. with ORGs.	13.81	9.78	Stdev LOCs	149.7	57.54
Avg. LOCs. per doc. with LOCs.	11.31	7.34	# docs. with 1 LOC	5,443	6,184
Avg. NEs per doc. with NEs	30.04	20.47	# docs. > 3 LOCs	12,913	11,640
Maximum # of LOCs in 1 doc.	20,594	6,472	# docs. > 30 LOCs	1,483	713

To investigate whether the kind of location occurring in Portuguese web texts had different properties: more fine-grained, or relating to physical geography (rivers, mountains, etc.), we looked into the subtypes of location NEs provided by SIEMÊS, shown in Table 2. The most frequent type of geographic NE is the name of a city, town or village (POV), followed by the name of a country. In fact, more than 85% of LOCs are concentrated in just three types (POV, ENDRALAR and SOCCUL) and the same occurs when counting only multi-word names.

Our explanation for this wealth of location NEs not present in Geo-Net-PT-01 (in addition to a systematic overgeneration of SIEMÊS, which will have to be analysed elsewhere, although we expect it to be of considerable import for the numbers presented here) resorts to the following hypotheses:

– given that Portugal is and has always been a small country with a very worldwide perspective, many (or even most) of the web pages do not concern only (or specially) Portugal – and so many geographical named entities concern places in foreign countries;
– in texts, people are bound to write about more fine-grained locations (often deictically), as “downtown”, “near my old school”, “in front of Jerónimos”, or “in the A1 highway”, which are not part of an administrative ontology.

Finally, with this study we also wanted to assess the hypothesis that location is a transversal semantic category, in the sense that geographical information can be found in (almost) all sorts of texts and not only in specialized technical texts (as for example those dealing with geography or tourism). So, we measured the total number of documents with at least one NE: 31,489 (98.4% of the snapshot). References to people are present in 21,499 (67.18%) documents, organizations in

30,328 (94.77%) documents and locations in 24,468 (76.46%) documents. Table 3 shows that each document (containing at least one NE) contains on average ca. 20 DNEs from which more than seven are localities and ca. 50% of the documents with LOCs contain more than three LOCs. The values of the "Distinct" column measure the distinct NEs into each document.

4 Concluding Remarks

We provide a first measure of geographical information (as far as named entities are concerned) in the Portuguese web, concluding that a geographic ontology built from web texts can complement administrative sources.

Conversely, and using the available frequency lists for the WPT 03 collection, we found out that ca. 20% of the one-word names in Geo-Net-PT01 do not occur in WPT 03 at all. This shows that – for indexing of Web contents – to know what the Web talks about may considerably reduce the index size, and provides another motivation for our work.

Acknowledgements

We are grateful to Mário Silva for pertinent comments and to Luís Sarmento for help with SIEMÊS. This work was supported by grant POSI/PLP/43931/2001 from Fundação para a Ciência e Tecnologia, co-financed by POSI.

References

1. A. Mikheev, M. Moens, and C. Grover. Named Entity Recognition without Gazetteers. In *Proc. of the Ninth International Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 1–8, Bergen, Norway, 1999.
2. Wenbo Zong, Dan Wu, Aixin Sun, Ee-Peng Lim, and Dion Hoe-Lian Goh. On assigning place names to geography related web pages. In *JCDL '05: Proc. of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries,*.
3. M. J. Silva, B. Martins, M. S. Chaves, N. Cardoso, and A. P. Afonso. Adding Geographic Scopes to Web Resources. *CEUS - Computers, Environment and Urban Systems, Elsevier Science*, 2006 (in press).
4. M. S. Chaves, M. J. Silva, and B. Martins. A Geographic Knowledge Base for Semantic Web Applications. In C. A. Heuser, editor, *Proc. of the 20^{th} Brazilian Symposium on Databases, Uberlândia, Minas Gerais, Brazil*, pages 40–54, October, 3–7 2005.
5. Luis Sarmento. SIEMÊS - a named entity recognizer for Portuguese relying on similarity rules. This volume.
6. Diana Santos, Nuno Seco, Nuno Cardoso, and Rui Vilela. HAREM: an Advanced NER Evaluation Contest for Portuguese. In *Proceedings of LREC'2006*, Genoa, Italy, 22-28 May 2006.

Corpus-Based Compositionality

Milena Garrão[1], Claudia Oliveira[2], Maria Claudia de Freitas[1],
and Maria Carmelita Dias[1]

[1] Pontifícia Universidade Católica do Rio de Janeiro,
Rua Marquês de São Vicente, 225, Rio de Janeiro, Brazil
[2] Instituto Militar de Engenharia,
Praça General Tibúrcio, 80, Rio de Janeiro, Brazil

Abstract. This paper introduces another mode of semantic compositionality, particularly regarding verbal Multi-word Expressions (MWE). Our hypothesis is that with the increase in similarity between the paragraphs containing a certain MWE and its separate atomic lexical items, the compositionality between these items inside the MWE also increases. The results are consistent with this hypothesis.

1 Introduction

In this paper we approach the concept of Multi-word Expressions (MWE) from a truly corpus-based perspective. By choosing this alternative path, we avoid the necessity to use time-consuming and controversial human intuition for the assessment of MWE-ness and we get much richer lexicographic information on the phenomenon, such as its statistics, and the behavior of its textual environments. In the "intuition framework", found in mainstream approaches to MWE description [4],the measure for distinguishing a MWE from a casual syntactic string is usually based on what is called compositional semantic measure. In other words, while an occasional syntactic string would be "semantically transparent", a MWE is often thought of as "semantically opaque" and therefore less compositional. This perspective often leads to controversial opinions and confusing criteria regarding MWE identification and delimitation. As an Alternative to these implications,our framework does not imply a priori word meaning and the MWEs' compositional measure is truly context-dependent.Thus, We can obtain more rapid and reliable results on MWE identification and description through a corpus-based approach.

The methodology employed, first tested on Brazilian Portuguese verbal phrases of the pattern V+NP, is a statistically-based corpus analysis which could be summed up as follows: 1) the use of a robust linguistic corpus as input; 2) the application of a filter to obtain V+NP patterns in the corpus; 3) the application of a probabilistic test to the results obtained by the filter, in order to raise the chances of obtaining MWEs rather than random syntactic combinations; 4) application of Similarity Measures between texts presenting the very same MWE. This latter step, inspired by Information Retrieval techniques [2], is vital to assess the level of compositionality of the MWEs. We empirically observed

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 268–271, 2006.

that with the increase in similarity between the paragraphs containing a certain MWE and its separate atomic lexical items, the compositionality between these items inside the MWE also increases.

2 Corpus-Based Compositionality: from Theory to Application

We share Kilgarriff's [6] perspective on the notion of word meaning. He states that meanings are purpose-dependent and argues that in the absence of such purposes, word senses do not exist. Therefore, we rely on a corpus-based framework. Our method can be presented in two phases: first, the most promising candidates expressions to MWE-ness are listed and ranked. The second step determines the compositionality level of the expression, based on an Information Retrieval technique that measures the similarities between the contextual adjacencies of the MWE. It is an essentially corpus-dependent compositional measure.In both phases we used CETENFolha (Corpus de Extractos de Textos Electrônicos NILC/Folha de S. Paulo)[1]. In addition, we designed a complementary small corpus, with the results of the very same pattern V+NP searched on the Internet for cases in which the number of occurrences of V+NP structures in CETENFolha was insufficient (less than 50 occurrences) for the reliable application of the test described above.

2.1 Spotting MWEs

This first phase in the method combines two different statistical steps: 1) the application of a filter to obtain V+NP patterns in the corpus; 2) the application of a probabilistic test to the results obtained (the Log Likelihood score) [7].

Step 1 is designed to pre-select instances of the pattern V+SN, deemed to be very frequent in Portuguese, as well as in other languages cf. [5]. Step 2 detects whether the co-occurrence is a casual syntactic string or if it is otherwise motivated (a MWE). This type of test requires two types of hypotheses, see [7]:

$$H_1 : P(w_1|w_2) = P(w_1|\neg w_2) \qquad H_2 : P(w_1|w_2) \neq P(w_1|w_2)$$

where, H = hypothesis, P = probability, w = word.

The hypothesis H_1 assumes that the two words are independent and hypothesis H_2 assumes otherwise. H_2 is the MWE hypothesis and the log Likelihood score measures how H_2 is more likely than H_1. For instance, if we assume that "fazer amor" is a MWE, it is expected that the independence hypothesis $H_1 : P(fazer|amor) = P(fazer|\neg amor)$ to be false. The NSP statistic set made possible the application of this test [3].

2.2 Measuring the MWEs Compositional Levels

After identifying the most common MWEs in the corpus, we observed the contrasts between the contexts in which they appear. This comparison is possible through the application of the Vector Space Model [2]. The two expressions whose semantic similarity we want to compute (for instance, “fazer campanha” and “campanha”) are represented as vectors in a multidimensional space. The co-sine between these vectors indicate the words they have in common, and thus it is taken as a measure of semantic similarity.

For each MWE w we carried out the following steps: i) extraction of all paragraphs containing w (set P_1); ii) extraction of all paragraphs containing the noun in w that does not occur in P_1 (set P_2); iii) indexing P_1 and P_2 in the Vector Space Model; iv) calculating the similarities matrix between paragraphs in P_1 and averaging its values; iv) calculating the similarities matrix between paragraphs in P_2 and averaging its values, v) calculating the similarities matrix between paragraphs in P_1 and P_2 and averaging its values.

Our hypothesis is that with the increase in similarities between the paragraphs in P_1 and the paragraphs in P_2, the level of compositionality of the expression V+SN also increases. In order to evaluate this hypothesis, we calculate the similarity intra-P_1, and then intra-P_2. Afterwards, we assess the similarity inter P_1 and P_2.

This method was applied to MWEs headed by the following verbs: “dar” (*to give*), “fazer” (*to make*),“receber” (*to receive*), “ter” (*to have*), “tomar”(*to take*), “ganhar” (*to get*, *to save*), “usar”(*to use*), “deixar” (*to leave*), “perder” (*to lose*). The following table is a sample of the quantitative results we obtained with each verb. It is organized as follows: *i.* the leftmost column contains the list of NPs, in the V+NP structure; *ii.* AS1 is the average similarity intra-P_1; AS2 is the average similarity intra-P_2; AS3 is the average similarity inter P_1 and P_2; Var are the corresponding variances.

Table 1. Results with verb *ter*

TER	AS1	Var	AS2	Var	AS3	Var
fôlego	1,066	0,01	2,17	0,005	0,34	0,0005
acesso	0,86	0,007	3,04	0,01	0,36	0,0002
uma idéia	0,54	0,008	1,94	0,009	0,36	0,0004
razão	1,00	0,007	5,29	0,12	0,42	0,001
sucesso	1,47	0,03	4,11	0,04	0,43	0,0007
fora	5,98	0,13	4,06	0,006	0,44	0,0004
problema	0,88	0,002	2,51	0,02	0,72	0,001
medo	1,28	0,017	5,42	0,07	0,84	0,006

2.3 Overall Analysis of the Results

Table 2 shows in the leftmost column the MWEs which were evaluated by our method as the most compositional ones. The right column displays the least compositional ones. The middle column displays the borderline cases.

The overall results are consistent with our predictions,thus grounding our inevitable intuitions about the Portuguese MWEs’ patterns of compositionality. On the other hand, there were some others which contradicted our intuitions. For instance, “tomar café” and“café” did not follow the same compositional pattern of “tomar banho” and ”banho”. In other words, the noun “café” without the

Table 2. Sumary of the results

	+ compositional	**Borderline cases**	**- compositional**
Fazer	fazer amor; fazer campanha	fazer sucesso	fazer falta;fazer sentido
Ter	ter medo; ter problema	ter força	ter fôlego;ter uma idéia
Dar	dar lucro; dar declaração	dar sorte	dar bandeira; dar frutos
Perder	perder emprego; perder a eleição	perder tempo	perder a cabeça; perder o bonde
Usar	usar camisinha; usar drogas	usar o cinto	usar a cabeça; usar a força
Receber	receber propina; receber benefício	receber visita	receber alta
Deixar	deixar vestígio; deixar filhos	deixar o país	deixar marcas
Tomar	tomar cuidado; tomar providência	tomar decisão	tomar partido; tomar iniciativa
Ganhar	ganhar eleição; ganhar o jogo	ganhar tempo	ganhar espaço; ganhar terreno

MWE structure appears in different semantic environments, such as economics, agriculture, design (as a color), etc. This has a great impact on the similarity measure results. This also means in our perspective that this noun is polysemic.

3 Concluding Remarks and Future Work

We are extremely confident with this empirical approach to a semantic measure. We are also aware that there are some adjustments to be made. These results could be even more reliable if there was a wider tagged corpus of Brazilian Portuguese texts. We could also reflect on improving the results by considering not only one paragraph but a larger linguistic scope for the purpose of comparison. But for now, we can say that there is a set of linguistic applications that could benefit from it: Information Retrieval, Machine Translation, not to mention lexicography as a whole. In fact, what we find more motivating is that Semantics benefits from it.

References

1. R. Aires and S. Aluisio. Criação de um corpus com 1.000.000 de palavras etiquetado morfossintaticamente. Technical Report 08, NILC, Sao Paulo, Brazil, 2001.
2. R. Baeza-Yates and B. R. Neto. *Modern Information Retrieval.* Addison Wesley, 1999.
3. Banerjee and Pedersen. The design, implementation, and use of the ngram statistic package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational linguistics*, pages 370–381, 2003.
4. M. Gross. Une classification des phrases figées en français. *Revue Québecoise de Linguistique*, pages 151–185, 1986.
5. F. Guenthner and X. Blanco. Multi-lexemic expressions: an overview. *Linguisticae Investigaciones Supplementa*, pages 201–218, 2004.
6. A. Kilgarriff. I dont believe in word senses. *Computer and the Humanities*, pages 91–113, 1997.
7. C. D. Manning and H. Schütze. *Foundations of statistical natural language processing.* The MIT Press, Cambridge, Massachusetts, 1999.

Author Index