UNIVERSITY OF WOLVERHAMPTON

SCHOOL OF LAW, SOCIAL SCIENCES AND COMMUNICATIONS

UNIVERSIDADE DO ALGARVE

FACULDADE DE CIÊNCIAS HUMANAS E SOCIAIS

A Supervised Machine Learning Method for Word Sense Disambiguation of Portuguese Nouns

Marcos Zampieri

A Project submitted as part of a program of study for the award of

MA Natural Language Processing & Human Language Technology

Dr. Constantin Orasan (University of Wolverhampton)

Dr. Jorge Baptista (Universidade do Algarve)

May 26th, 2010

UNIVERSITY OF WOLVERHAMPTON

SCHOOL OF LAW, SOCIAL SCIENCES AND COMMUNICATIONS

MA NATURAL LANGUAGE PROCESSING & HUMAN LANGUAGE TECHNOLOGY

Name: Marcos Eduardo Zampieri de Marco

Date: May, 14th, 2010

Title: A Supervised Machine Learning Method for Word Sense Disambiguation of Portuguese Nouns

Module Code: LN4007

Presented in partial fulfilment of the assessment requirements for the above award.

Supervisor: Dr. Constantin Orasan

Declaration:

This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

It is acknowledged that the author of any project work shall own the copyright. However, by submitting such copyright work for assessment, the author grants to the University a perpetual royalty-free licence to do all or any of those things referred to in section 16(i) of the Copyright Designs and Patents Act 1988 (viz: to copy work; to issue copies to the public; to perform or show or play the work in public; to broadcast the work or to make adaptation of the work.

This project did not involve contact with human subjects, and hence did not require approval from the LSSC Ethics Committee.

Signed:_____ Date:_____

ABSTRACT

Word Sense Disambiguation (WSD) is vital in many Natural Language Processing (NLP) applications. This work aims to explore supervised machine learning techniques for the disambiguation of Portuguese nouns. The primary motivation for this work was the conceptualization of WSD integrated in an Information Retrieval (IR) engine in order to show how WSD may improve document retrieval from the world-wide web. After a brief overview of the most relevant applications for WSD, the main approaches and state-of-the-art techniques available for the task are presented. For the comparison of different WSD algorithms and techniques, a selection of ambiguous words from a Portuguese academic vocabulary was taken and a catalogue of word senses was established for each of them. A training corpus of real occurrences of each word in context was collected, providing manually annotated contextual data for each sense of the ambiguous word. The corpus was processed and features were extracted using Python and the Natural Language Tool Kit (NLTK) to feed into machine learning algorithms. Results are evaluated and discussed.

RESUMO

Desambiguação lexical de sentido, do inglês Word Sense Disambiguation (WSD), é uma tarefa vital para muitas aplicações na área do Processamento de Linguagem Natural. O trabalho aqui apresentado visa explorar técnicas de aprendizado supervisionado para a desambiguação de substantivos em Português. A motivação principal desse trabalho surgiu da ideia de integrar técnicas de desambiguação lexical a um motor de busca para recuperação de informação e com isso, mostrar como um módulo de desambiguação automática pode aumentar a performance da recuperação de documentos da internet. Após uma breve introdução acerca das aplicações mais relevantes de WSD, as principais técnicas para a resolução da tarefa serão discutidas.Com intuito de estudar diferentes algoritmos e técnicas para desambiguação automática, foi efetuada a seleção de palavras ambíguas a partir de um vocabulário académico do Português e um corpus de ocorrências reais de cada palavra em contexto foi coletado. O corpus foi processado utilizando a linguagem de programação Python e os componentes do NLTK e diferentes algoritmos foram utilizados. Ao fim, resultados são avaliados e discutidos.

ACKNOWLEDGMENT

I am very thankful to my supervisors, Dr. Constantin Orasan at the University of Wolverhampton and Dr. Jorge Baptista at the University of Algarve, whose support and encouragement made it possible for me to successfully complete this dissertation.

I would also like to thank Dr. Lucia Specia, for her contribution as a lecturer in Computational Linguistics and for her technical help in the early stages of the experiments carried out in this work.

I am also grateful to the coordinator and the partner universities of the Erasmus Mundus Masters in Natural Language Processing and Human Language Technologies (EM NLP – HLT). They selected me among candidates from all over the world for the Erasmus Mundus scholarship award that funded this master degree.

I am thankful to all my colleagues in the NLP-HLT Masters for everything we have experienced, lived and learned together in these two years of Masters. Three colleagues have helped me in a special way in this dissertation and I am very thankful to them, they are: Simone Pereira, who collaborated as an annotator in the Kappa experiments; Binyam Gebre who provided help and useful hints about machine learning techniques and finally, Alice Kaiser-Schatzlein who proofread this dissertation as an English native speaker.

I would also like to thank the mobility office staff at the University of Algarve, specially Marlene and Paula Simões. They receive students from all over the world and in spite of all cultural differences, do their best to make students feel at home. I was one of them.

Agradeço também a meus pais, que com muito esforço e dedicação lutaram para que eu pudesse receber uma educação de boa qualidade e me tornar a pessoa que sou hoje. Sempre, em todos os momentos, acreditaram no meu potencial de aluno e ser humano. Tenham certeza de que apesar da distância, continuarei fazendo o possível para que o orgulho que sentem, seja sempre maior do que a nossa saudade.

As previously mentioned, this work was funded by an Erasmus Mundus scholarship offered by the European Union Education and Training Commission, EMMC 2008-0083 at the Erasmus Mundus Masters in NLP & HLT.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Ambiguity is intrinsic to human language and it constitutes an important challenge for most computational applications in the field of Natural Language Processing (NLP). Ideally, systems should be able to deal with ambiguity in order to increase performance in NLP applications such as Machine Translation, Text Summarization and Information Retrieval.

The process of automatically assigning the correct sense of ambiguous words in context is called Word Sense Disambiguation (WSD) and for the last few decades, WSD has constituted a well-established independent area of study within the NLP community.

Given the relevance of WSD in state-of-the-art NLP applications, this work presents a supervised machine learning method for the automatic disambiguation of nouns in Portuguese. The work aims to compare currently used machine learning approaches and algorithms described in the literature and it also discusses the application of WSD in an Information Retrieval search engine. The content of this dissertation is organized as follows:

The first chapter presents a brief description of the WSD task as well as its main applications. The sample application that has motivated this work is discussed in more detail in section 1.3. It consists of automatic disambiguation rules to be integrated in an Information Retrieval search engine in the scope of REAP.PT, a Computer Aided Language Learning (CALL) software. Section 1.4 presents an overview of the main approaches for WSD described in the literature divided in two subsections: Early Approaches and Corpora and Machine Learning.

The second chapter introduces the key concepts of the computational techniques used in this work, namely machine learning. A brief overview of the most used algorithms, features, input and output types are presented. The use of machine learning in different NLP tasks is also explored in this chapter in section 2.4.

All the methods used for the experiments are presented in the third chapter. It begins with the selection of ambiguous words based on an academic vocabulary. After the selection of words, the methodology behind the compilation of the training corpus is presented in section 3.2. The issue of inter-annotator agreement and the methodology to calculate the baseline for each word is also presented in this chapter in sections 3.3. and 3.4. Pre-processing of the corpus data and feature extraction are also explained in the last two sections of the third chapter

The fourth chapter presents the results obtained for the Most Frequent Sense (MFS) baseline and also Inter-Annotator Agreement. The results were obtained using three different machine learning classifiers in Python NLTK, namely Naïve Bayes, Decision Tree and Maximum Entropy. These classifiers are presented and discussed in section 2.3.

In the fifth and last chapter, the conclusions and further perspectives for research in the field of WSD are presented as well as some considerations about the integration of the automatic disambiguation rules into an Information Retrieval engines and more specifically the REAP.PT.

Two documents are included as appendix to this dissertation. The first one is a sample of real occurrences of the word *arquivo* extracted from the corpus. The second one is a sample of the Python program used for the disambiguation of the word *arquivo*.

CHAPTER 1 - Word Sense Disambiguation, Applications and Approaches

This first chapter is concerned with presenting an overview of the Word Sense Disambiguation (WSD) task, its main applications in the field of Natural language Processing (NLP) and the main approaches described in the literature to address this problem.

The first section, section 1.1, this section will present a definition of the task itself. Since WSD is a task that addresses lexical ambiguity and the meaning of words, a brief discussion about word meaning, lexical semantics and relations between meanings will be held in section 1.2.

In section 1.3., the main applications that benefit from WSD will be presented such as Machine Translation, Text Classification, Question Answering and Information Retrieval. The latter will be discussed in more detail in section 1.4, using a practical application, the REAP.PT, as an example.

Finally, the chapter ends with section 1.4, where the main approaches used for the WSD task are described with respect to the state-of-the-art. In section 1.4, and also subsection 1.4.1, there is a brief description of the main conference in the field, SEMEVAL.

1.1. Word Sense Disambiguation Task

Many words have more than one meaning in natural language, and each one of them is determined by its context. For example, the Portuguese word *apêndice* (*appendix* in English) is defined in commonly used dictionaries such as Houaiss for Brazilian Portuguese and Porto Editora for European Portuguese as:

1. (book part) A separate part at the end of a book or magazine which gives additional information to the readers;

2. (body part) A small tube-shape part which is joined to the intestines;

Both senses of *apêndice*, book part or body part in their respective context, are easy to recognize for any Portuguese native or competent speaker. However, for NLP applications, this distinction is not always trivial and can generate problems in language processing.

The automated process of deciding word senses in context is known in NLP as WSD. Research in WSD has increased in recent years in an attempt to increase performance in several language processing tasks, however its need had already been detected in early NLP applications:

> "WSD has been a research area in Natural Language Processing for almost as long as the field has existed. It was identified as a distinct task in relation to machine translation (MT) over forty years ago. One of the first problems which was faced by these systems was that of word-sense ambiguity. It became rapidly apparent that semantic disambiguation at the lexical level was necessary for even trivial language understanding." (Stevenson and Wilks, 2003: 250)

WSD is not an NLP application on its own, but an important component which has brought significantly increased performance when incorporated in several systems. The main applications for WSD will be presented in section 1.3, and before that, in section 1.2, a brief discussion about meaning and relations among meaning, in particular, homonyms and polysemes are presented.

1.2. Ambiguity and Sense Distinction

As described in the previous section, ambiguity is part of human language and it is a linguistic phenomenon that has been a topic of study in linguistics for many years, especially in the field of semantics. Semantics is commonly defined as the study of meaning, however, some linguists, such as Kearns (2000), claim that the study of semantics covers only a part of meaning, referring to the literal one:

> "The study of linguistic meaning is generally divided in practice into two main fields, semantics and pragmatics. Semantics deals with the literal meaning of words and the meaning of the way they are combined, which take together form the core of meaning, or the starting point from which the whole meaning of a particular utterance is constructed. Pragmatics deals with all the ways in literal meaning must be refined, enriched or extended to arrive at an understanding of what a speaker meant in uttering a particular expression." (Kearns, 2000: 1)

When dealing with ambiguity in practical computational applications such as the one described in this work, research is delimited to lexical ambiguity. In other words, the learning methods are only concerned with literal meanings of words that can in most cases asserted by context. Further derivations of meaning, such as the ones studied by pragmatics, constitute a much more complex task for NLP systems: as stated by Leech and Weisser (2000), and their need is usually restricted to spoken dialogue systems (SDS).

As the scope of this work is restricted to lexical meaning, it is important to point out that the study of meaning in the lexicon constitutes an important sub-area in semantics, usually referred to as lexical semantics. Pustejovski (1995), and Pustejovski and Boguraev (1996), discussed in detail the fundamental questions in lexical semantics and lexical meaning, such as polysemy, sense extension and lexical disambiguation.

There have several attempts to explain why ambiguous words appear in human language, and to describe how ambiguity is manifested. One of the fundamental works in this area is the one written by the American linguist George Kingsley Zipf (1949), in which proposed several empirical laws based on the *Principle of Least Effort* to phenomena such as ambiguity. In one of the

laws, Zipf claims that word sense ambiguity arises as a result of two competing forces, both of which try to follow the principle of least effort. On one side, the speaker´s effort is minimized by using a small vocabulary, in other words, by using few words that are capable of conveying many meanings. On the other side, the listener's effort is minimized by using as many distinct words (in terms of meaning) as possible.

The laws proposed by Zipf are widely accepted in the linguistic community, especially because of the application of the empirical evidence provided by statistical measures to linguistic data to demonstrate empirical evidences in it. This was innovative for human language studies in the late 1940´s, however in current literature, there are studies exploring other aspects of meaning, ambiguity and human language as a whole.

Ambiguous words can be categorized in two large groups, polysemes or homonyms. Homonyms are commonly used to refer to two senses of a word that do not present any obvious semantic relation. For instance, *ball,* which can be a dance or a round shaped object. Etymology is usually considered for homonyms, so that senses with the same historical origin are grouped together.

In a polysemic relation, it is usually assumed that the words have the same origin. This is the case of the word *branch,* which as a noun has the following definitions: 1. one of the parts of a tree that grows out from the main trunk and has leaves, flowers or fruit on it; 2. a part of something larger; 3. one of the offices or groups that form part of a large business organization; 4. a part of a river or road that leaves the main part.

In the case of *branch*, the senses described by the definitions presented above hold a clear semantic relation between them. Therefore, meanings for the word *branch* can be grouped as being part of a whole structure, either a tree, an organization or a river.

Following this brief introduction to word meaning and relations between them, this work will avoid further philosophical discussions about meaning and relations between word meanings, since those discussions can turn out to be misleading in practical computational tasks such as WSD. The dissertation will

focus solely on the practical task of automatic disambiguation and will address the sense distinctions using available linguistic resources such as lexicons. Native speakers´ knowledge will also be taken into account and measured through an estimation computed by the Kappa coefficient to quantify between annotators.

In the next section the main applications for WSD will be briefly presented.

1.3. WSD main applications

The WSD task is an important component of several NLP systems, such as Machine Translation (MT), Question Answering (QA) Information Retrieval (IR), Information Extraction (IE), Text Classification and several speech processing applications.

Researchers in MT have concentrated efforts on WSD since the earliest NLP applications (Stevenson, 2003). MT researchers identified that their results would be considerably increased by using WSD methods to disambiguate words in automatic translation for various pairs of languages, such as English and Portuguese (Specia, 2007).

Lexical ambiguity in MT systems can occur in the source or target language. In some cases, a word can be ambiguous in both languages, as is the case with the pair *appendix* in English and *apêndice* in Portuguese, where the same two senses, book part and body part, can be found in both languages. In other cases, such as the pair *bank* in English and *banco* in Portuguese, the most common sense of the words is equal, meaning financial institution. However, in English *bank* also means the land along the side of a river, which in Portuguese should rather be translated as *margem*.

In Portuguese, *banco* means a seat which could be translated to English as *bench*. There is, a sense of *banco* in Portuguese related to the English *bank* of the river, which is described as a small island of sand and stone in the middle of the river, but the use is rare. Those are clear examples of how lexical

ambiguity presents itself in machine translation systems and how the number of ambiguities to be dealt with increases compared to a monolingual system.

In Question Answering applications, WSD is useful to link question words to answer words. When users pose questions to a QA system it is very likely that the question will contain at least one ambiguous word. Therefore it is necessary for the system to decode the question with the correct sense of the ambiguous word according to context in order to search for the correct answer in the database.

In some aspects, WSD for QA applications is very similar to the next application to be presented: Information Retrieval. When answering a question, QA systems perform what is described as paragraph extraction to present the correct answer to the user and at this stage it works with a full question instead of just keywords as in IR. However, in some cases the question may be too short, and it may not contain the necessary contextual information for disambiguation.

Information Retrieval is another language processing application that benefits from WSD. Most of the words used to execute queries in IR systems have more than one meaning and therefore when performing a query the system may retrieve documents which are not relevant to the search (Kulkarni et al. 2008).

As a real life example using a popular search engine on the internet, if a user types the word *bank* it is very likely that the first results the person will get will be related to the financial institution. However, a particular user may not be interested in searching *bank* as a financial institution, but the other sense, *bank* of a river. In this case then this user would probably search again, this time adding related words before or after the main word *river*, *water* or *grass*.

In this very simplistic example, it is possible to understand how information about the context of word senses, in an IR engine, can help the queries to be more accurate. A sample application for IR, will be presented in more detail in the next section.

State-of-the-art research in the use of WSD for Text Classification and Text Categorization systems presents an interesting paradigm. WSD was considered for many years to be a task that would improve performance when applied before text classification. The idea behind it was that once the correct senses of ambiguous words in text were assigned, it would be less likely that ambiguity would mislead text classification.

Gomes Hidalgo et al (2005) explored the use of WSD prior to the task of Automatic Text Categorization (ATC), which consists of automatic assignment of documents to pre-defined categories. ATC is considered to be one of the most important tasks in Text Classification and like other tasks in the domain; it takes advantage of WSD to improve performance.

In recent years, some researchers have looked at the problem from the opposite perspective and claimed that domain information could be used in the WSD task itself:

> "Magnini et. al (2002) have shown that information about the domain of a document is very useful for WSD. This is because many concepts are specific to particular domains, and for many words their most likely meaning in context is strongly correlated to the domain of the document they appear in." (Koeling, McCarthy and Carroll 2007)

In the example of *appendix*, text classification could be applied and if a text containing this word is identified as belonging to the medical domain, it is very likely that the word refers to the body part rather than the book part.

Another potential application of WSD takes place in the field of speech processing. Some words in natural language, called homophones, are pronounced the same, but they have different spellings and therefore different meanings, for example: *rain* and *reign*, both pronounced /reɪn/. In other cases, the words are homographs, they are written the same, but their pronunciation changes according to the meaning, f example: *bass* can be a music instrument pronounced /beɪs/ or a fish pronounced /bæs/.

In speech recognition systems, a WSD module can be used to increase systems´ performance by distinguishing senses for homophone and homograph

words according to their context. A speech synthesis application may also benefit from WSD and generate more accurate and natural pronunciation as discussed by Yarowsky (1996b and 1997).

In the next section, the possibilities of a WSD module in an information retrieval application will be explored in more depth using the example of the REAP.PT, Computer Aided Language Learning (CALL) software.

1.3.1. Sample Application - WSD for IR - REAP.PT

The initial idea of this dissertation emerged from an application described in the previous section: Information Retrieval. More precisely, a disambiguation module is considered necessary to increase performance of an IR engine developed to be part of a computer-aided language learning (CALL) software, the REAP.PT. (Marujo, 2009).

REAP.PT is the Portuguese version of REAP and it is currently in development by an interdisciplinary research group in Portugal, with engineers from the Spoken Languages Systems Lab. of INESC-ID in Lisbon and also linguists from the University of Algarve and Lisbon.

The acronym REAP stands for "**Rea**der-Specific Lexical **P**ractice for Improved Reading Comprehension". It is a tool for computer-aided language learning (CALL) developed at the Language Technologies Institute (LTI) of Carnegie Mellon University (CMU). (Collins-Thompson and Callan, 2004)

REAP aims to improve language skills and vocabulary learning for foreign learners of English. Its main task is to retrieve texts from the internet according to specific criteria and the students´ preferences. After retrieval, the texts are presented to the students along with reading-practice exercises to help them acquire new vocabulary or new contexts for words already known.

Texts are retrieved according to each student´s level of proficiency. Students can also select areas of interest in their profile by rating 16 general topics including Arts, Science and Sports, according to their preferences.

Students´ preferences along with their respective level of proficiency constitute the student´s profile, which will help the search engine to retrieve texts that are relevant to each student. This process makes the reading practice more stimulating and suitable to the students´ learning goals. The REAP.PT IR module also has general predefined filters to avoid offensive language.

REAP.PT is also integrated with an online Portuguese dictionary, the *Dicionário da Língua Portuguesa* developed by Porto Editora, as well as a speech synthesis application which can read the whole text or specific parts of it, as defined by the student.

As shown previously, a WSD module can improve the accuracy of Information Retrieval systems. In the case of REAP.PT, a WSD module can diminish the probability of the system retrieving impertinent information because the system´s search mechanisms will be more likely to retrieve texts that contain a word´s desired sense.

The REAP research team at Carnegie Mellon University (CMU) is currently carrying out experiments in WSD for the English language (Kulkarni et al. 2008), but so far, the WSD module has not been integrated. These experiments were published in 2008 and used supervised machine learning techniques for the disambiguation of 30 nouns in English.

Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB) were the algorithms used for the task and the results varied for each word, in a range of 86.27% to 99.82% accuracy. The experiments used two groups of features: unigrams and part-of-speech bigrams. Unigrams are lexical features, unique words that occur in the training set in a specific window range, whereas POS bigrams are lexico-syntactic features combining POS information in a predefined range of the target word.

The study went beyond the disambiguation task and evaluated the disambiguation methods through students´ performance in the classroom, performing reading practice experiments. The results were described as follows:

> "It is demonstrated that the task of disambiguating homonyms can be automated by learning classifiers that can assign the appropriate sense to a homonym in a given context with high accuracy. A user study reveals that students equipped with WSD-enabled vocabulary tutor perform significantly better than students using vocabulary tutor without the WSD capabilities." (Kulkarni et al., 2008)

As the REAP is currently being developed for Portuguese, the aim of this dissertation is to replicate the first experiments described by Kulkarni, et. al (2008) for the disambiguation of Portuguese nouns. Therefore, experiments in presented this dissertation will be concentrated on the disambiguation task itself rather than its integration to REAP.PT. In so doing, this work will constitute the foundations for the development of a wider disambiguation module for use in the Portuguese version of REAP.

In the next section a brief overview of the main approaches described in the literature about the state-of-the-art for automatic disambiguation will be presented. There will be a subsection on the comparison of approaches done by conferences, such as SENSEVAL (which later became SEMEVAL).

1.4. Approaches to WSD

This section presents an overview of the main approaches for WSD divided into two subsections. The first one describes the early approaches to address the task of automatic disambiguation, which started as a part of MT translation and as an independent task that relied on dictionary definitions found in Machine Readable Dictionaries (MRD) and rule-based algorithms.

The second subsection presents state-of-the-art applications, most of which use data extracted from corpora combined with machine learning algorithms and statistical measures.

1.4.1. Early Approaches

The first need for disambiguation emerged from MT systems, thus the very first approaches to WSD considered the task as part of the analysis module of MT systems. WSD was therefore not considered as a topic of study on its own. The first attempts to address lexical ambiguity as an autonomous task occurred in the 1980´s and today it constitutes an important area of research in NLP and Computational Linguistics.

> "Of course, WSD is not thought of as an end in itself, but as an enabler for other tasks and applications of computational linguistics and natural language processing (NLP) such as parsing, semantic interpretation, machine translation, information retrieval, text mining, and (lexical) knowledge acquisition." (Agirre, Edmonds, 2006: 2-3)

Among the first studies of automatic disambiguation on its own, Hirst (1987) aimed to provide an abstract semantic representation of the entire input text, making it possible to distinguish senses of ambiguous words in the text. Even though conceptually lexical ambiguity could be resolved by semantic representation, further studies have shown that this kind of approach aims too high due to what is described in the literature as the knowledge acquisition bottleneck.

The knowledge acquisition bottleneck is a problem not only in WSD systems, but also in other software applications. Software engineers take it as the main challenge for the design and implementation of any knowledge based system. In the case of WSD, it is necessary to define which sources of data are relevant and provide enough information for the task prior to data acquisition.

Reviews of the state-of-the-art such as Stevenson (2003) and Jurafsky and Martin (2009: 671 - 714), usually present several approaches separately and make very fine distinctions between them according to techniques, resources and algorithms used. In fact, there are three fundamental issues that may be considered in order to classify different WSD approaches: data sources, the algorithm and additional linguistic resources.

Data source for non-ambiguous contexts can be found in Machine Readable Dictionaries (MRD) such as the Longman Dictionary of contemporary English (LDOCE) (Procter, 1978). In the 1980´s dictionary publishers started to develop electronic versions of their dictionaries, and this solved one of the bottlenecks of the early WSD approaches, the coverage of lexicons. However, while those dictionaries had the necessary lexical coverage for robust WSD systems, they also contained a large amount of polysemy, which meant a higher number of potential senses for each word.

> "(polysemy in MRD) led researchers to attempt to disambiguate texts relative to the senses found in MRDs as a way of utilizing these knowledge sources." (Stevenson, 2003:12)

In the example of *bank*, the Cambridge dictionary presents five different senses for the word bank as a noun, some with fine distinctions between them. Because of the high level of polysemy present in machine readable dictionaries, after some years of research, MRD were not considered an optimal source of knowledge for WSD and other resources were also employed (as discussed later in this section).

Lesk (1986) was one of the first researchers that tried to disambiguate MRD definitions using algorithms His algorithm became well-known among WSD researchers. Hence, several researchers tried to adapt the algorithm to obtain better performance including the work of Banerjee and Pedersen (2002), which used WordNet[1] for English (Miller et. al., 1993).

The Lesk algorithm is based on the assumption that words in a given neighbourhood will tend to share a common topic, and therefore it aims to disambiguate words in short phrases. Given an ambiguous word, the dictionary definition of each of its senses is compared to the definitions of every other word in the sentence. The algorithm assigns the word sense whose definition shares the largest number of words in common with the definitions of the other words. The algorithm begins a new process for each new word and does not use the senses it previously assigned.

[1] http://wordnet.princeton.edu/

Lesk demonstrates the application of the algorithm on the words *pine* and *cone*. Using the Oxford Advanced Learner's Dictionary, the word *pine* has two senses:

- Sense 1: kind of **evergreen tree** with needle–shaped leaves
- Sense 2: waste away through sorrow or illness.

For the word *cone* three senses can be found:

- Sense 1: solid body which narrows to a point
- Sense 2: something of this shape whether solid or hollow
- Sense 3: fruit of certain **evergreen tree**

Each of the two senses of the word *pine* is compared with each of the three senses of the word *cone* and it is found that the phrase *evergreen tree* occurs in one sense for each of the two words. These two senses were then declared to be the most appropriate senses when the words *pine* and *cone* are used together in a sentence.

Lesk claimed that his early approach correctly disambiguated 50% to 70% of words from some short samples of the Jane Austen novel, Pride and Prejudice.

Another source of data for WSD can be found in electronic corpora, as discussed in the next section. The availability of large bodies of text in recent years has made it possible for researchers to search for patterns in the data, through statistical models or machine learning applications.

1.4.2. Corpora and Machine Learning

One of the pioneer studies on corpus usage in WSD was detailed in the paper by Ng and Lee (1996). In this approach, called "exemplar-based learning", the word sense was assigned to the sense of the most similar example already seen by the system. This approach is considered to be a

supervised learning approach which requires previously disambiguated training text.

Algorithms for WSD can rely on rules to assert the correct sense of a word however this kind of approach is not as widely used in state-of-the-art applications as it was in early approaches. The use of statistical methods and machine learning techniques has significantly increased in the last few years.

In machine learning, there are two ways of learning: supervised and unsupervised. Supervised learning relies on a set of previously established senses, whereas in unsupervised learning, those classes are not pre-set and therefore the algorithm must perform clustering of similar classes prior to disambiguation. Several works describe the use of machine learning algorithms to the word sense disambiguation task, such as the previously mentioned Kulkarni et al. (2008), as well as Yarowski, (1996a) and Florian et. al. (2002).

A wide variety of features can be used in WSD approaches using machine learning algorithms. A brief overview about these features, as well as a more detailed description of the features used in this work can be found in chapter 3, from section 3.7 onwards.

It is possible to address WSD using a combination of two knowledge sources (dictionary and corpora), which constitutes a hybrid approach, as described by Luk (1995). Corpora to be used in WSD may be untagged (also called raw corpora) or previously tagged. Tags can vary according to the information to be added to the data and studies show that the performance of WSD algorithms can, in some cases, be improved by the addition of linguistic information to raw data. Examples of linguistic resources used in WSD are POS tags, syntactic information obtained by syntactic parsing tools, semantic tags, and more recently ontologies such as Wordnet.

Ontologies usually contain information about semantic relations between words. Therefore hyponym, hypernym and meronym relations for words can be automatically assessed, which ideally constitutes a substantial gain in semantic information. However, the use of ontologies also faces some of the same restrictions as WSD using dictionary definitions, most importantly the high level

of polysemy in these resources. Specia (2007) described the use of definitions from Wordnet for the disambiguation of verbs in Portuguese in a MT system.

In this work, a corpus-based approach was used for supervised machine learning. It is considered to be a supervised approach because a catalogue of senses was established prior to the training stage and the senses were marked in the data. This corpus data was processed using a feature extractor in Python NLTK, which was subsequently used by machine learning algorithms.

In order to compare different approaches fairly, it is necessary to have well established criteria and evaluation methods. Moreover, the data should be equavalent or similar to avoid bias in evaluation. In this context, the role of evaluation campaigns and conferences for WSD is discussed in the next section.

1.4.3. Comparing approaches

Evaluation will be described in the fourth chapter of this dissertation and it plays an important role in WSD systems and is by no means a trivial task. Results may vary due to data distribution, mainly because WSD often receives unbalanced data. Unbalanced data means that one sense is more frequent than the others and therefore there will be more examples of it than the other senses.

The most relevant WSD evaluation campaign for the international NLP community is SENSEVAL[2], today called SEMEVAL. The first SENSEVAL conference took place in England in 1998 and it evaluated systems that worked with English, French and Italian. The second edition occurred in 2001 in Toulouse, France and there was an increase in the number of languages from three to eleven languages, establishing SENSEVAL as the main conference in WSD and related applications.

[2] http://www.senseval.org/

Twelve years after the first edition, this year´s SEMEVAL-2 presents 18 different tasks in several languages with a broader scope than the previous editions. The tasks range from Coreference Resolution to Sentiment Analysis in adjectives, reference resolution of temporal expressions, disambiguation of compounds, etc.

These conferences play an important role in the evaluation of state-of-the-art systems in WSD and they help to establish current benchmarks. They also provide a set of standardized training and testing data that is reliable for comparing approaches. Conferences also propose metrics for evaluation that can be used commonly between systems.

In this work, training and test data were split using the technique of n-fold cross validation, widely used in evaluation campaigns such as SENSEVAL, SEMEVAL and described in section 2.5. The evaluation metrics, namely accuracy, precision, recall and f-measure were the same used to evaluate in SEMEVAL conferences, as described in section 4.4.

CHAPTER 2 - Machine Learning and its Applications in NLP

In this chapter, the key concepts of machine learning will be presented as well as its main applications, particularly in the field of Natural Language Processing.

The chapter begins in section 2.1., which presents the scope and usefulness of machine learning in state-of-the art computational applications. In section 2.2 and its subsection, the main concepts of machine learning are presented, namely the input and output of these applications. Section 2.3, is dedicated to the explanation of three of the most important algorithms that were used in the experiments of this dissertation, Naïve Bayes, Decision Trees and Maximum Entropy.

Section 2.4 presents the most important applications of machine learning in NLP, such as POS Tagging, Syntactic Parsing and Information Extraction.

Finally, the chapter ends with a brief discussion of evaluation techniques, focusing on n-fold cross validation, the strategy that was used in the experiments presented in this work.

2.1. Scope and Usefulness

As briefly mentioned in section 1.3, Machine Learning and statistical measures are often used in state-of-the-art NLP applications. In the last decade, the NLP community has observed a research paradigm shift from rule-based approaches to statistical and machine learning approaches.

This change was observed in a wide range of applications in NLP such as Machine Translation, pre-processing tasks such as POS tagging (Marquez et al., 2000) and the task described here, Word Sense Disambiguation (Specia, 2007), (Kulkarni et al., 2008). A reasonable number of tools for NLP researchers developed in the past few years contain plug-ins and integration to ML toolkits, such as MALLET (McCallum, 2002) and the Natural Language Toolkit (NLTK) (Bird, Klein and Loper, 2009) which will be used for the experiments described here. More about the use of machine learning in NLP tasks will be discussed in section 2.4.

Rule-based approaches continue to be used and described in the literature however their usage is currently limited to the domain of the application. The use of hybrid approaches where statistical measures are combined to rules has also increased in the NLP community in recent years.

Machine Learning involves a computer algorithm learning from data. Based on a set of predefined features, algorithms identify patterns in data and can therefore infer predictions. There are however some philosophical discussions how learning is defined. To avoid misunderstanding, the following definition suits the scope of the machine learning task developed in this dissertation:

> "Things learn when they change their behavior in a way that makes them perform better in the future." (Witten, Frank, 2005:8)

In current literature the words "learn" and "train" have been used interchangeably. However, "training" in some cases tends to be more appropriate for computational application, since "learning" has an intelligent component that training does not.

Machine learning is the overarching concept and the practical task of learning patters from existing data is described in the literature as data mining. In some cases, both terms are used, in NLP the term machine learning is usually used whereas in Information Technology and Business Intelligence applications the term data mining is widely used. However,

2.2. Key Concepts

To have a good understanding of the whole process of machine learning and its practical application, it is very important to understand three fundamental issues: the input, the output and the algorithms used for learning. Understanding what the input and outputs are is usually more important than what goes in between.

It is important to have an intuition of what each machine learning algorithm do in practice, and one should be conscious of its advantages and disadvantages. Understanding machine learning customization is a far more complicated task and it is usually carried out by experienced programmers in the field. These researchers are usually more interested in the potential and results of algorithms themselves rather than their use in practical tasks, like the one presented here.

According to Witten and Frank (2005) there are basically four styles of learning or concepts to be learned within ML applications, as follows:

- Classification learning: A supervised model which is provided the actual outcome or class for each of the training examples. Classification algorithms arrange instances according to these predefined classes, in other words they aim to predict the class of each instance.

- Association: Differs from classification, since it searches for relations between variables, usually in large data sets. Methods in association

learning try to search for patterns in data in which two or more variables combined are likely to provide a given result.

- Clustering: An unsupervised method, where the classes are not previously defined. The algorithm groups items together automatically, according to similarities found on data.

- Numeric prediction: Generates an outcome that is a numeric quantity rather than a discrete class, unlike the three previous styles of learning.

The task developed in this work was a supervised classification task and is represented as figure 1:

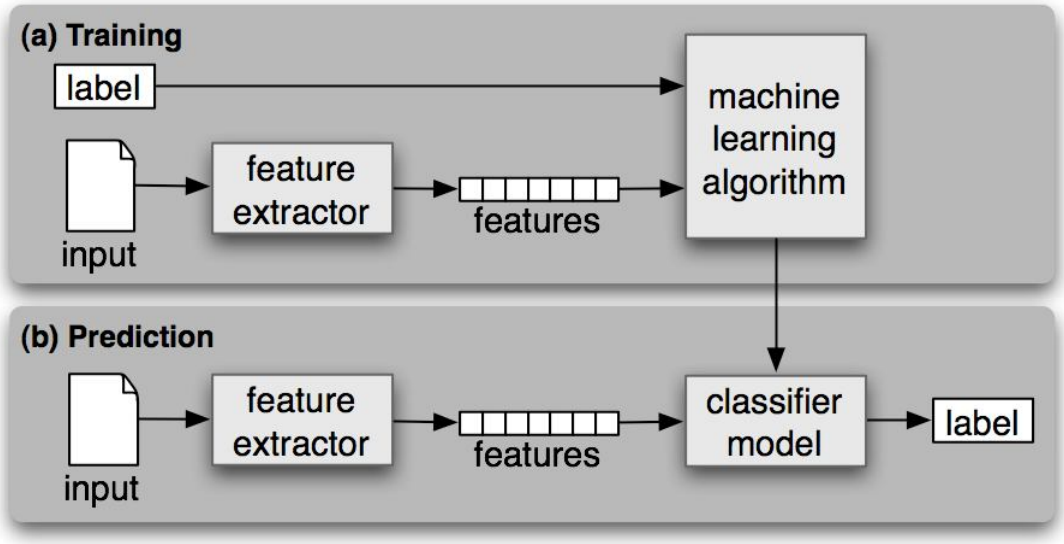

Figure 2.1: Training in classification problem (Bird, Klein and Loper, 2009)

The model represented in Figure 1 is the sample scheme of a supervised classification task using a machine learning approach. It is divided in two boxes (a) training and (b) prediction, also called testing. In box (a) the input data is prepared using a feature extractor, a set of instructions, usually a script or a program, to convert raw data into features. In the case of the experiments presented in this work, corpus concordances were prepared as instances

containing features and attributes by a feature extractor, using a Python program, that will be described in the methods section 3.3.

On the top, the set of possible labels are defined beforehand, therefore constituting a supervised approach. These labels are also provided to the machine learning algorithm. Machine Learning algorithms apply learning methods to the features and labels provided to create a model of classification that can be used for new or previously unseen data.

Once the model is created, the label prediction presented in (b) is applied, using a sample of the same dataset as the training dataset and the same features, in order to ensure a fair evaluation of the model. The classifier model will determine, based on the features and attributes, which class the new test instance belongs to. The output generated is usually the labeled data for error analysis. After analysis, further improvements can be made to the training features and therefore the classification model.

Continuing in key concepts, the next subsections will present the forms of inputs and outputs in machine learning models.

2.2.1. Input

The input in machine learning takes the form of concepts, instances and attributes, whereas the model to be learned is called concept description. The information given to the learner is a set of instances. Each instance is an individual and independent example of the concept to be learned and each of them is characterized by the values of a set of predetermined attributes. These attributes measure different aspects of the instance and the values presented can be numeric, nominal, binary, etc.

In a logical sequence, the definitions discussed so far can be organized as follows:

1. Concept Description – general concept to be learned
2. Concepts – individual units to be learned

3. Instances – examples extracted from the data organized with attributes

4. Attributes – predefined values for each instance

In classification learning, a concept is presented with a set of classified examples from which it is expected to learn to classify unseen examples. Classification learning is described in the literature as a supervised method, because it operates by being provided with the actual outcome of the training examples, also called the class of the example.

To feed machine learning algorithms, each dataset is represented as a matrix of instances versus attributes, which in database terms is a single relation. Problems arise when preparing data, mostly when the data involves relationships between objects rather than being separated as independent instances.

The input data can be arranged in various ways including tables and hierarchical trees. Most machine learning tools available, such as WEKA (Witten and Frank, 2005) work with input in the form of a text file. WEKA uses a comma separated value (CSV) file, used mostly for digital storage of data structured in a table of lists form, where each associated item (member) in a group is in association with others also separated by the commas of its set.

For this reason it is necessary to develop a program or script to prepare the data before using any of the tools available. For the experiments presented in this work, Python NLTK was used to write a feature extractor to prepare the data.

In the next subsection some of the forms of the output of machine learning systems will be presented.

2.2.2. Output

The output of a machine learning system is also known as knowledge representation. They are patterns that are discovered or learned from the data

by different machine learning methods. They can be presented and schematized in many ways.

A brief description of the most important types of outputs is described in the next subsections. For this description, a fictitious example entitled “the weather problem” (Witten and Frank, 2005: 10 - 11) will be used. The dataset contains examples of weather conditions that are suitable for playing or not some unspecific game.

2.2.2.1. Decision Table

The simplest and most rudimentary way of representing the output from machine learning is a decision table:

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes

Table 2.1: Sample decision table for the weather data (Witten and Frank, 2005: 15)

The four leftmost columns: *outlook, temperature, humidity and windy* are parameters that were extracted from the original input data, whereas the fifth column *play* represents the predicted class from each of the instances, with the values *yes* or *no*.

2.2.2.2. Decision Tree

The same example used in the last subsection, the weather problem by Witten and Frank (2005: 15), is presented next in the form of a decision tree output.

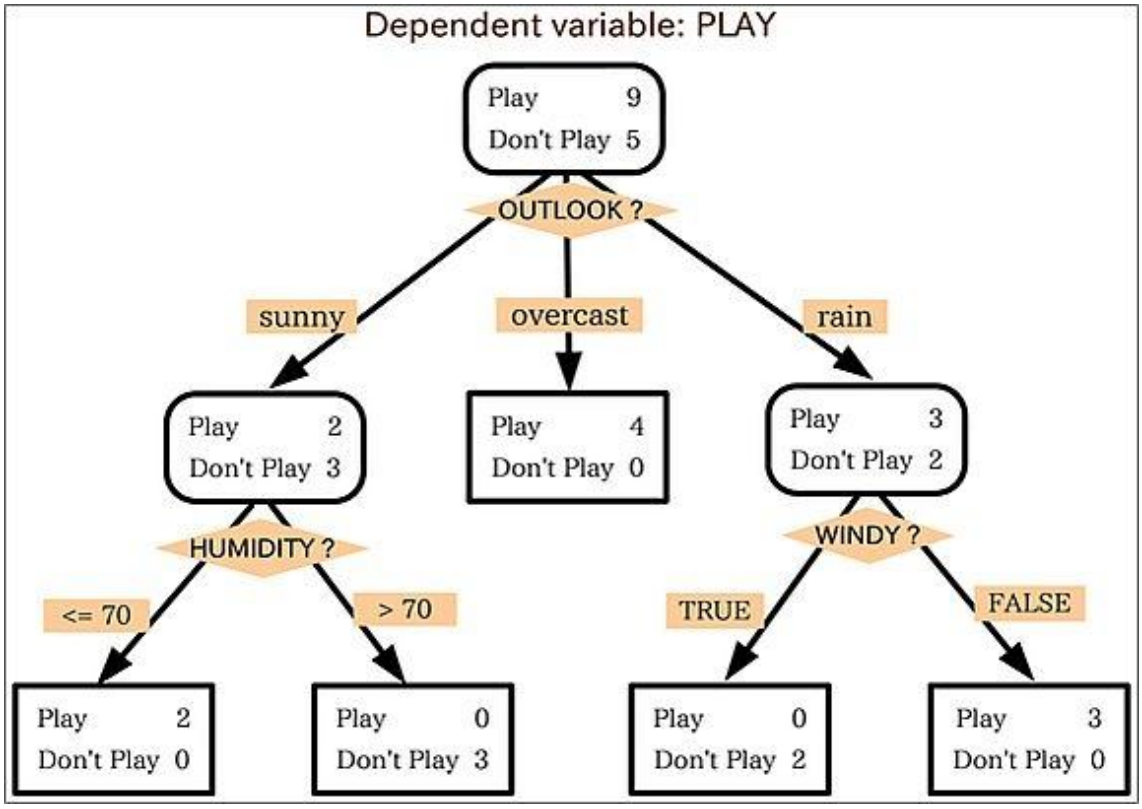

Figure 2.2: Decision tree output, the weather problem.

Decision Trees as an output are particularly useful when dealing with a limited number of variables, such as the weather problem. When dealing with a larger number of variables, it is usually difficult for one to analyze the tree in order to understand the predictions made by the algorithm.

Beginning on top, the tree represents 14 instances, labeled as *play* and *don´t play*. According to the variables, the tree expands and shows the expected labels according to each of the variables. In the second level, for example, the tree presents different probabilities regarding the two classes, if the weather outlook is *sunny*, *rainy* or *overcast*.

2.2.2.3. Classification rules

Classification rules are a popular alternative to decision trees. It is structured with an antecedent and a consequent (also known as a conclusion)

that gives the class or classes that apply to instances covered by that rule. A short sample using the weather problem is presented to illustrate:

```
If outlook = sunny and humidity = high then play = no

If outlook= rainy and windy = true then play = no

If outlook = overcast then play = yes
```

2.2.2.4. Association rules

The last type of output presented here are the association rules. They are derived from the classification rules, but in this case they can predict any attribute and not just the class, as the example bellow:

```
If temperature = cool then humidity = normal
```

Given that the experiments presented in this work will discuss performance and accuracy rather than knowledge representation for WSD, from the next section on, this chapter will be focused on the algorithms rather than explaining more in depth systems´ output.

2.3. Algorithms

Provided that machine learning methods are being widely used in state-of-the-art computational applications, not only in NLP, a wide number of algorithms have arisen. Popular algorithms in machine learning include: Naïve Bayes or Bayesian Classifiers, Support Vector Machines, Mutual Information, Decision Trees, Regression trees, Maximum Entropy and Condition Random Fields.

In the next subsections the three algorithms used for the experiments in this work Decision Trees, Naïve Bayes and Maximum Entropy, will be described in more detail. These algorithms were chosen firstly because they are popular methods in WSD, frequently used in state-of-the-art applications. Secondly,

they were chosen because their implementations are available in Python NLTK, making it easier to use them for further adaptations.

2.3.1. Decision Trees

Algorithms for constructing decision trees are among the most well known and widely used of all machine learning methods. Decision Trees are also widely used for classification, their accuracy is competitive with other learning methods and they are also very efficient. The learned classification model is represented as a tree.

Decision trees have been used as classifiers for numerous real-world domains, e.g., medical diagnosis and credit approval. For many of these domains, the trees produced by decision tree algorithms are both small and accurate, resulting in fast, reliable classifiers. These properties make decision trees a valuable and popular tool for classification.

Decision tree learning is typically done using the divide-and-conquer strategy that recursively partitions the data to produce the tree. At the beginning, all the examples are at the root. As the tree grows, the examples are sub-divided recursively. The leaves of the tree are demonstrated as presented in the pseudocode by Finlay and Dix, (1996):

```
buildtree ( examples: example_set ) returns node_ptr
  IF examples all of the same class
  THEN  return a pointer to a LEAF where the class is the one with the
                most examples and the certainty factor is the number
                in the class divided by the total number of examples.
  ELSE
      choose criteria C
      IF choosing process fails
      THEN return pointer to LEAF as above
      ELSE
        split_ex (C, examples, yes_examples {VAR}, no_examples {VAR})
        LET yes_b = buildtree( yes_examples )
```

```
        AND no_b  = buildtree( no_examples )
        return a pointer to a BRANCH
                        with criteria   =  C
                             yes_branch =  yes_b
                             no_branch  =  no_b
     END {if choosing fails}
  END {if examples of same class}
```

Figure 2.3: Pseudocode decision tree. (Finlay and Dix, 1996)

There are several implementations of decision tree algorithms described in the literature: one of the most famous and widely used implementations of it is the C4.5 algorithm, proposed by Quinlan (1992). The C4.5 algorithm is a successor of the IDE3 algorithm also invented by Quinlan (1986).

2.3.2. Naïve Bayes Classifier

The Naïve Bayes classifier produces probability estimates rather than predictions. This algorithm was first proposed by Duda and Hart (1973) and is based on Bayes´ theorem, also known as Bayes´ Law or Bayes´ Rules. The Naïve Bayes Classifier is a simple probabilistic model with strong naïve independence assumptions, described in the literature as an “independent feature model”.

The key idea of the Bayes´ theorem is that the probability of an event A given an event B depends not only on the relationship between A and B but on the absolute probability, or occurrence, of A not concerning B, as well as the absolute probability of B not concerning A. Bayes theorem can be formalized and represented by the following formula:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Where P(*A*) is the prior probability or marginal probability of *A*. P(*A*|*B*) is the conditional probability of *A*, given *B*, also called the posterior probability because it is derived from or depends upon the specified value of *B*. P(*B*|*A*) is the conditional probability of *B* given *A*, also called the likelihood. P(*B*) is the prior or marginal probability of *B*, and it acts as a normalizing constant.

In simple terms, a Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties as independent contributors to the final probability.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning dataset. In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations, leading researchers such as Zhang (2004) to analyze Bayesian classifiers in order to point out reasons how a naïve approach can perform classification tasks such with high performance:

> "Naive Bayes is one of the most efficient and effective inductive learning algorithms for machine learning and data mining. Its competitive performance in classification is surprising, because the conditional independence assumption on which it is based, is rarely true in realworld applications. An open question is: what is the true reason for the surprisingly good performance of naive Bayes in classification?" (Zhang, 2004)

An advantage of the Naive Bayes classifier is that it requires only a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined, and not the entire covariance matrix.

Naïve Bayes is a popular method in WSD and has been shown significantly good results in spite of its naïve approach.

2.3.3. Maximum Entropy

The Maximum Entropy model, also referred to as ME or Maxent for short, is a general purpose machine learning framework that has been successfully applied in various fields of research including spatial physics, computer vision, and also Natural Language Processing. In the experiments presented in this dissertation, the classifier available in NLTK was used to perform WSD.

Maximum entropy is a general technique for estimating probability distributions from data. The basic principle in Maximum Entropy is that when nothing is known, the distribution should be as uniform as possible, in other words, present the maximal entropy. In a classification task, labeled training data is used to derive a set of constraints for the model that characterize the class-specific expectations for the distribution. Constraints are represented as expected values of features or any real-valued function of an example.

There are several applications of Maximum Entropy in NLP. The work by Berger, Pietra and Pietra (1996) proposes the use of Maximum Entropy in an English - French MT system. Nigam, Lafferty and McCallum (1999) use Maxent for text classification and Ratnaparkhi (1996) uses a maximum entropy model for part-of-speech tagging.

2.4. Machine Learning Applications in NLP

As previously mentioned, machine learning is not only used in NLP and Customer Relationship Management (CRM) applications such as the one described in the previous section also benefits from the use of ML techniques. Companies spend a substantial amount of money to predict customer´s habits resulting in the acquisition of a great amount of structured data:

> “Some of the most active applications of data mining have been in the area of marketing and sales. These are domains in which companies possess massive volumes of precisely recorded data, data which – it has only recently been realized – is potentially extremely valuable. In these applications, predictions themselves are the chief interest: the structure of how decisions are made is often completely irrelevant.” (Witten, Frank, 2005:26)

Prior to releasing a new product and therefore a marketing and advertisement campaign, a company might want to know if its customers are likely to accept and to buy this new product. If they are ready to accept the new product, data mining techniques can be applied in the company´s sales database to infer which set or which group of customers would be interested in it. Therefore a marketing strategy can be planned to target those customers maximizing the investment in advertisement and sales.

Other areas might also benefit from ML techniques in order to obtain reliable predictions and this is the case of diagnosis. Medical diagnosis and device maintenance diagnosis can benefit from the aid of expert systems, each one with its own properties.

In the medical domain, based on the clinical history of a given patient, his or her symptoms and a database of records of other patients, expert systems might give doctors an insight into the diagnosis of a given illness. For electromechanical devices, applications based on historical information about failure and maintenance, and also general occurrences regarding the item, could infer a pattern among events and predict when a preventive maintenance should be done in order to avoid failure of the equipment.

In Natural Language Processing, there are also several applications using machine learning with satisfactory results, such as part-of-speech tagging. POS tagging is one of the most important applications in NLP and is usually considered one of the vital steps of text pre-processing. POS tagging can use machine learning algorithms and achieve high results as described by Brill (1995) and later by Marquez, Padro and Rodrigues (1999).

To determine the POS of a word the features used are usually the POS tags of the neighbouring words, two to three on both sides. For known words it is possible to use a dictionary to provide the set of possible POS categories for the word, whereas for unknown words all POS categories are in theory acceptable. To increase the likelihood of asserting a correct POS tag, especially in unknown words, morphological features can be used as additional input

features. POS taggers for English report accuracy above 96 per cent in state-of-the-art applications.

Another suitable machine learning application in NLP are syntactic and semantic parsing. One of the first applications of machine learning techniques for the task of syntactic parsing was employed to parse the Wall Street Journal (WSJ) of the Penn Treebank (Marcus, Santorini and Marcinkiewicz, 1993). The model applied for the task was the one described by Magerman (1995) and defined as statistical decision trees. The paper reported precision and recall of 84 per cent.

For semantic parsing, the lack of large annotated corpora with detailed semantic representations constitutes a bottleneck for the use of machine learning methods in domain independent applications. However, for domain specific applications, examples were used to learn patterns from data such as the work described by Ng and Zelle (1997). Ng and Zelle used machine learning for semantic interpretation by focusing on two tasks: Semantic Parsing and WSD.

Information Extraction (IE) can also benefit from the use of machine learning techniques. IE is a research area that is concerned with the extraction of particular information of texts. Its importance has significantly increased in the last decades due to the huge amount of information available online on the internet. Given its complexity, IE is usually used in domain-specific applications.

IE is a well-suited application for machine learning because these systems require a significant amount of knowledge of the texts´ domain. Further, building a rule based approach for IE is a very laborious task which can provide only limited performance and coverage. A number of rule induction methods have been applied to learning patterns for IE, such as Freitag (1998). In this approach, the training examples from texts are paired with filled templates and these systems learn patter-matching rules for extracting the appropriate units to fill slots.

More recently, Anaphora and co-reference resolution systems have used machine learning techniques with satisfactory results. Resolving anaphora or

identifying multiple phrases that refer to the same entity is a difficult language processing problem as described by Mitkov (2000). Several papers describe the use of machine learning techniques to address anaphora and co-reference resolution.

Aone and Bennet (2000) described methods to build an automatically trainable anaphora resolution system. In this approach, the authors used Japanese newspaper articles tagged with discourse information as training examples for a machine learning algorithm which employs the C4.5 decision tree algorithm.

After discussing the main applications from WSD in the field of NLP, the next section will discuss evaluation strategies for machine learning. This section will focus on a particular type of data distribution, called cross-validation.

2.5. Evaluation - Cross-validation

Evaluation plays an important role to determine the accuracy of any learning method. As was previously mentioned, any machine learning method needs data for training and test. There are several ways of doing it and the most common is to split data into two sets, usually 70% for training and 30% for testing or in some cases 80% for training and 20% for testing.

Although this distribution is commonly used for large datasets, it presents a challenge for smaller datasets and it might lead to problem of representativeness of the training or testing data. Therefore it is necessary to ensure that random sampling is done in a way that guarantees that each class in the data set is properly represented in both the training and test sets.

To avoid inaccuracy of results due to data splitting, a statistical technique called cross-validation can be applied. In cross-validation, a fixed number (n) of folds or partitions of the data are assigned, and it is referred to as n-fold cross-validation. In the case of a three-fold cross-validation, data is split into three equal partitions, two of them used for training and the last one is used for testing. The process is repeated three times to ensure that all the instances in

the data set were used to train and to test. For evaluation purposes, the average of the three iterations is calculated.

In order to predict the error rate of a learning technique given a fixed sample of data, the use of 10-fold cross-validation has become common in the machine learning research community:

> "Extensive tests on numerous datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up." (Witten and Frank, 2005: 150)

For these experiments, the method of n-fold cross validation is used divided in three sets, each set containing 33% of the total data, therefore a three-fold cross validation. The three-fold was chosen mainly because the amount of data used for the experiments is not considered to be big as in most other applications. Because of that, fewer partitions were employed in order to ensure that a reasonable number of instances are included in each partition.

CHAPTER 3 - Methods

In this chapter, the methods used in the experiments will be presented. It begins with the selection of ambiguous words from a Portuguese academic vocabulary, the Portuguese Academic Word List P-AWL (Baptista et al., 2010), The selection and sense distinctions for these words were made according to linguistic criteria discussed in section 3.1.

Based on a random sample of 100 occurrences of each word, the Most Frequent Sense (MFS) baseline was established and it is presented in section 3.1. Along with the MFS baseline, these sample occurrences were used to calculate a Kappa coefficient in order to measure inter-annotator agreement, in section 3.4.

After the selection of words and senses, and the establishment of the baseline and inter-annotator agreement, chapter 3 presents the processing of data, namely: tokenization and feature extraction. The three groups of features used will be described in section 3.7. and its subsections.

3.1. Word selection and Catalogue of Senses

As the main motivation for this research is restricted to academic texts, the selection of the vocabulary used in this task was based on academic vocabulary. For this selection the Portuguese Academic Word List, P-AWL, was used as the first resource.

> "P-AWL is a general purpose vocabulary, with current (but not colloquial) words, which has been designed for immediate application on a CALL web-based environment, currently devoted to improve students' reading practice and vocabulary acquisition." (Baptista et al., 2010).

P-AWL contains 1823 words in total and it is the Portuguese equivalent to the Academic Word List (Coxhead, 2000) compiled for English.

In table 3.1, the POS distribution of P-AWL is shown:

POS	Count
Noun	754
Adjective	451
Verb	409
Adverb	203
Conjunction	4
Preposition	2
Total	**1823**

Table 3.1: Breakdown of P-AWL by POS (Baptista et. al, 2010)

The academic list was scrutinized in a word by word analysis to compile a list of 13 ambiguous nouns with two major senses that are clearly distinguishable for any Portuguese native or competent speaker. It also took into account the possible differences of word sense between the Brazilian and European Portuguese.

Nouns were chosen because they constitute the largest POS category in every corpus of Portuguese and most other languages. They are also often ambiguous and its ambiguity appears in different domains and contexts. As is shown in table 3.1, nouns are also the most frequent category among the entries of the P-AWL, which was used as the reference for vocabulary selection

It is common for a word to manifest ambiguity in two different parts of speech, as is the case of *call* which has twelve senses described by the Cambridge Dictionary, five as a noun and seven times as the verb *to call*. However, ambiguity between two words with two different POS, were not considered for these experiments. This distinction between ambiguity in the same or in different POS was described by Yarowsky (1996a):

> "Most of the ambiguities studied and evaluated…focus on the sense ambiguities within the same part of speech (such as distinguishing the two noun senses of crane), while leaving the distinction between noun and verb sense of the word (e.g. to crane your neck) as the responsibility of the part of speech tagger." (Yarowsky, 1996a:8)

Thus some ambiguities are resolved at the level of the POS tagging. State-of-the-art results for POS taggers are currently above 95%. For the sample application in this dissertation, the REAP.PT, WSD will be in future integrated in an NLP chain (Mamede, 2010), which features a POS tagger that reports above 97% precision.

Commonly-used dictionaries such as Porto Editora for European Portuguese and Houaiss and Michaelis for Brazilian Portuguese were used as a repository of the senses. When choosing the words, two linguists were directly involved one Portuguese and other Brazilian. Other four researchers, three linguists and a computational linguist also provided feedback about ambiguity.

The two major senses, according to corpus frequency and native speakers´ linguistic intuition, formed the two main classes for the experiments, Sense 1 and Sense 2. The remaining occurrences of the words, like compounds, proper nouns and minor senses (when existent) formed a third class, called Sense X. With this approach, every occurrence of a word was taken into account in the training and test data and classified as 1, 2 or X.

The list is composed of the following words are sense definitions (their respective English translation in brackets):

- apêndice
    1. Parte acessória de um órgão. (Accessory part of an organ.)

2. Em uma obra, anexo que a complementa; acréscimo, suplemento. (In a written production, the annex that complements it, supplement.)

- arquivo
  1. Conjunto de documentos manuscritos, gráficos, fotográficos etc. produzidos, recebidos e acumulados no decurso das atividades de uma entidade pública ou privada. (Set of documents, manuscripts, graphs or photographs, etc. Produced received and accumulated during the activities of a public or private entity.)
  2. Conjunto de dados digitalizados que pode ser gravado em um dispositivo de armazenamento e tratado como ente único. (Set of digital data that can be recorded in a computational storage device and treated as a unique piece.)
- comissão
  1. Conjunto de indivíduos que uma assembléia incumbe de executar determinada tarefa especial, realizar um estudo, examinar e opinar sobre um negócio, resolver problemas etc. (Group of individuals that an assembly charges with execute a given task, study or examination and forming an opinion about an issue or to solve a problem, etc.)
  2. Percentagem ou prêmio que representantes comerciais, caixeiros-viajantes, corretores, vendedores etc. cobram sobre o valor dos negócios realizados ou sobre o produto do trabalho prestado. (Percentage or bonus that sales people, retailers, etc. charge against the total value of a given business, product or work done.)
- crédito
  1. Confiança, crença alimentada pelas qualidades de uma pessoa ou coisa; segurança de que alguém ou algo é capaz ou veraz. (Confidence, faith based on the qualities of a certain people or thing; confidence that something or someone is capable or truthful.)
  2. Acordo ou contrato pelo qual um estabelecimento bancário, uma financeira ou afim põe determinada quantia à disposição de alguém mediante assinatura de letras de câmbio, notas promissórias ou qualquer outra prova de formalização do compromisso. (Agreement or contract in which a bank, a financial institution or a related organization offers certain amount of money to someone provided that this person presents signatures or any other form of legal commitment.)
- cultura
  1. Conjunto de padrões de comportamento, crenças, conhecimentos, costumes etc. que distinguem um grupo social. (Set of standarts of behaviour, belief, knowledge, costums, etc. that distinguish a social group.)
  2. Ação, processo ou efeito de cultivar a terra; lavra, cultivo. (Act, process or effect of cultivating the land.)

- essência
    1. Aquilo que é o mais básico, o mais central, a mais importante característica de um ser ou de algo, que lhe confere uma identidade, um caráter distintivo. (The most basic, most central and most important characteristic of something or someone, that attributes it na identity and a distinguishable feature.)
    2. Óleo fino e aromático, extraído por destilação de flores, folhas, frutos ou raízes de certos vegetais. (Aromatic oil extracted through the destilation of flowers, leaves, fruits or roots of certain vegetables.)
- etiqueta
    1. Rótulo, letreiro, adesivo etc. em que se podem identificar algumas características e/ou informações referentes ao objeto que os contém. (Label, sign, tag, etc. in which one can identify some characteristics and/or information about the object that it contains.)
    2. Conjunto de regras de conduta, especialmente as de tratamento, seguidas em ocasiões geralmente formais, e que revelam sobretudo a importância social das pessoas envolvidas. (Set of rules of behaviour, especially for treating people, usually followed in formal ocasions that reveal the social importance of the people involved.)
- foco
    1. Ponto central de onde provém ou para onde converge alguma coisa; centro. (The central point from where something comes or to where it converges; centre.)
    2. Rubrica Óptica: Ponto para o qual converge, ou do qual diverge, um feixe de raios luminosos paralelos, após a reflexão por um espelho esférico. (Optical domain: Point to which something converges or diverges, paralel raylights, after the reflexion of it by an spherical mirror.)
- garantia
    1. Ato ou efeito de garantir(-se). Ato ou palavra com que se assegura o cumprimento de obrigação, compromisso, promessa etc. (Act or effect of guarantee. Act or word that one insures the fulfillment of an obligation, commitment, promise, etc.)
    2. Documento que assegura a integridade de um produto vendido e/ou a boa qualidade ou durabilidade de um serviço prestado, e que obriga o fabricante a consertar ou substituir a mercadoria com defeito e o prestador de serviço a refazê-lo se insatisfatório. (Document that insures the integrity of a sold product and/or its good quality of service that obliges the manufacturer to repair or replace the product or the service provider to redo the service, if the result is not satisfactory.)

- geração
    1. Espaço de tempo correspondente ao intervalo que separa cada um dos graus de uma filiação e que é avaliado em cerca de 25 anos. (Range of time that corresponds to the interval that separates each stage of ancestry that is estimated to be around 25 years.)
    2. Ação ou efeito de gerar(-se). (Act or effect of generating.)
- imagem
    1. Gravura, representação da forma ou do aspecto de ser ou objeto por meios artísticos e/ou tecnológicos. (Ilustration or representation of the form or aspect of an object by artistic and/or technological means.)
    2. Opinião (contra ou a favor) que o público pode ter de uma instituição, organização, personalidade de renome, marca, produto etc.; conceito que uma pessoa goza junto a outrem. (Opinion (favorable or not) that the public can have about and institution, organization, personality, brand, product, etc..)
- regime
    1. Ação ou maneira de reger, de dirigir, de governar. (Action or manners of ruling, conducting, managing and governing.)
    2. Conjunto de prescrições qualitativas e quantitativas concernentes aos alimentos destinados a manter ou a restabelecer a saúde, ou a provocar o emagrecimento ou o aumento de peso; dieta. (A set of qualitative and quantitative prescription of aliment in order to maintain or reestablish the health, or to lose or gain weight; diet.)
- volume
    1. Bibliologia: Cada uma das partes brochadas ou encadernadas separadamente de uma obra impressa, conjunto dos números de um periódico publicados. (Bibliology: Each of the binding parts of a given printed production, book. Set of documents of published periodicals.
    2. Magnitude de um corpo, um objeto, uma edificação; corpulência, grandeza. (The extent of a body, an object, a construction; largeness, greatness.)

In chapter 4 a table containing the frequency of each selected word in the corpus is presented, as well as a calculation of its frequency in proportion to the amount of words in the corpus.

3.2. Training and Test Corpora

The NILC corpus available at Linguateca[3] (Santos, 2000 and Santos et. al., 2004) was used to collect the examples for the training corpus. Queries in the database were made for each word in the wordlist in order to capture all of the occurrences in the corpus. The queries were made using simple regular expressions in the corpus interface.

For the selection of the corpus sentences, not all the occurrences of the words were considered and only a predefined set of forms were taken into account. Nouns in their basic form (no diminutives, augmentatives or superlatives) were selected.

For each sense of the target word, a set of sample sentences was selected in the corpus in order to create the training corpus. For this selection, the following criteria were taken into account:

- Only full sentences were selected and not fragmentary text;
- Titles, list items, legends, and other paratextual elements were not collected;
- Definitions and other lexicographic or meta-linguistic context were avoided;
- The target word did not appear at the beginning nor at the end of the sentence;
- Examples were considered with a length varying between 100 to 200 characters;
- Examples corresponded to a ‘natural’ or ‘characteristic’ distribution of the target word;
- In principle, they should also constitute a non-ambiguous environment for the correct identification of an ambiguous word.

These criteria were established to ensure that all the examples or instances, for training and testing had the necessary length, clear context and

[3] http://www.linguateca.pt/acesso/corpus.php?corpus=SAOCARLOS

distribution to be classified by the algorithms. For example short sentences provide usually less information for automatic disambiguation. The position of the target word is also important and in the case of the word appearing in the beginning or in the end of the sentence, features such as neighbouring words will have less information to assert the correct sense of a word.

The importance behind this selection is that the analysis of the results can be concentrated on the methods itself rather than on the quality of the examples provided for training and testing.

3.3. Baseline

The baseline for this task was established using the notion of Most Frequent Sense (MFS). MFS baseline is a simplistic approach in which for a set of occurrences containing the ambiguous word, the most frequent sense is assigned.

The MFS baseline is considered an ideal method to establish the bottom line for WSD methods, because in order words, it is equivalent to no disambiguation. It is used as the standard baseline for most works in WSD, including conferences such as SENSEVAL.

To calculate the MFS baseline for this set of words, a step was done manually by looking at the occurrences extracted from the corpus and classifying each of them into one of the established senses. The number of occurrences for the establishment of the baseline was set to 100, and they were sorted to avoid any bias.

Results for the MFS baselines are presented in chapter 4.

3.4. Inter-Annotator Agreement

WSD is more than a laborious computational challenge of automatically asserting the right sense of a given word in a context. WSD proves to be also a

matter of disagreement when it comes to what are the actual senses of the words. Commonly-used dictionaries often present a very high level of polysemy due to very fine sense distinctions.

The idea of inter-annotator agreement is to measure how well native and competent speakers can agree on a given meaning of word in context. Comparison between annotators regarding sense distinction provides information on how difficult established senses are to distinguish.

NLP applications such as POS tagging or syntactic parsing usually present a high level of agreement between different annotators. This means that after defining precise criteria for the annotation, there is usually no disagreement.

However, the agreement for sense distinction is not as easy to achieve. There are two reasons for this:

> "Firstly, it is rarely the case that any two dictionaries will have the same set of sense definitions for a given word. Different dictionaries tend to carve up the "semantic space" in a different way, so to speak. Secondly the list of senses for a word in a typical dictionary tends to be rather refined and comprehensive. This is especially so for the commonly used words which have a large number of sense." (Ng et al., 1999)

The broader coverage of dictionaries tends to lead researchers to define a large amount of senses for a given ambiguous word. In consequence to that, not only the automatic disambiguation task becomes more difficult but the agreement between human annotators tends to decrease substantially.

In this task the number of senses established was restricted to those clearly distinguishable for any Portuguese native or competent speaker, and only two major senses or classes were established.

The experiment consisted of providing the description of the senses along with 100 random occurrences of that given word in context extracted from the corpus. These occurrences were given to different Portuguese native speakers, who were asked to assign only one sense.

Inter-annotator agreement was usually calculated using the Cohen´s Kappa index:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

In the formula presented above, P(A) represents the proportion of times that the annotators agreed and P(E) represents the proportion of times annotators were likely to agree by chance.

Results from the Kappa index vary between 0 and 1. Carletta (1996) pointed out that there is no standard interpretation for the results obtained, however it is considered acceptable to use the following scale:

- 0 - 0.2 slight agreement among annotators
- 0.2 - 0.4 fair agreement among annotators
- 0.4 - 0.6 moderate agreement among annotators
- 0.6 - 0.8 substantial agreement among annotators
- Above 0.8 high agreement among annotators

Some considerations might be pointed out when talking about inter-annotator agreement in WSD. Most of them to exemplify how difficult it is to assign one sense, especially when the dealing with short sentences. The Portuguese word *geração* that was been used in the experiments can be translated to English as *generation* and has its roots in the Latin *generatione.* The two senses used here for disambiguation are:

1. All the people of about the same age within a society or within a particular family, a period of about 25 years.
2. Act of generating something.

Given these two senses, the following sentence is given:

> *par=27920: Uma nova **geração** de Mireyas. (A new **generation** of Mireyas.)*

The sentence above requires very specific contextual information in order to assert the correct sense of the word *geração*. One can argue by looking at the two definitions and not knowing the word *Mireya,* that this word, beginning with a capital letter, could mean anything. Therefore, it is feasible to believe in the "act of generating Mireyas." However, in Brazil it is a given name for women, not as usual as other given names, but still used, and not used in European Portuguese.

Words that occur frequently together, can present two meanings depending to its context. The words *queimar* and *arquivo* can be translated to English as *burn* and *file* respectively. In a regular distribution they have their literal meaning preserved, which is to set fire in a file. However, in Brazilian Portuguese *queima de arquivo* turned out to be a compound that means a specific type of crime of murdering someone that has valuable or compromising information about another person.

Examples are presented to illustrate:

> *O oficial fugiu pelos telhados, e a população queimou **arquivos**, destruiu as balanças e libertou os presos.* (The official ran in the roof and the population burnt **files**, destroyed scales and freed the prisoners.)
>
> *Ele disse desconhecer que os bicheiros tenham sequestrado testemunhas da Procuradoria Geral de Justiça e que estejam fazendo queima de **arquivo**.* (He claimed to know that the criminals had kidnapped witnesses of the Justice Department and that they are doing "**file** burnt".)

In chapter 4 the results for inter-annotator agreement using the Kappa coefficient are presented.

3.5. Python NLTK

For the development of the experiments presented in this dissertation, the Python programming language in its version 2.6.2 for Windows was used. Python is an open source general-purpose high-level programming language

whose design philosophy emphasizes code readability. It combines remarkable power with very clear syntax and its standard library is large and comprehensive. Like other dynamic languages, Python is often used as a scripting language, but it is also used in a wide range of non-scripting contexts. Python is compatible with all major operating systems: Windows, Linux, Unix, Mac, OS/2, etc.

Python is widely used in the NLP community along with the Natural Language Tool Kit (NLTK). NLTK is an open source suite of libraries and programs for symbolic and statistical NLP using Python. It includes graphical demonstrations and sample data and it is accompanied by extensive documentation, including a book (Bird et. al., 2009). The book is available online and it explains the underlying concepts behind the language processing tasks supported by the toolkit.

3.6. Pre-Processing

Pre-processing is usually defined as a preliminary set of tasks used to prepare data for text processing. Pre-processing includes tasks such as tokenization, sentence splitting and lemmatization.

In the experiments presented here, pre-processing is the preparation of data extracted from corpus into a compatible format for use by machine learning algorithms in Python - NLTK. Since the sentences in the corpus were already split when they were extracted from corpus concordances, the main pre-processing task for these experiments was tokenization.

Tokenization is the process of identifying individual units in a text as words. It is a preliminary task to most text processing applications and for the features described in these experiments it is necessary to have a tokenized text.

The tokenization program for this task was coded in Python and it takes into account some specifications of Portuguese as presented in figure 3.2.

```
def tokenize(text):
      pattern = re.compile('([^!"#$%\'\s(),/:;?]+)')
      potential_tokens = pattern.split(text)
      for i in range(len(potential_tokens)):
            potential_tokens[i] = potential_tokens[i].strip()
      tokens = filter(lambda x: x != '',potential_tokens)
      return tokens
```

Figure 3.1: Python tokenizer for Portuguese.

After the pre-processing of the texts using the tokenizer, programs for feature extraction were written in Python and more about them will be presented in the next section.

3.7. Feature Extraction from Corpora

To prepare the data and the parameters to feed machine learning algorithms it is necessary to use a script or a program to convert raw data into training instances and extract predefined features.

A wide range of features can be used in WSD. In particular, *collocational* features that specify words which can appear in specific locations before and after the target word. Usually, this is set to a pre-defined window of two, sometimes three words on each side. Binary features are also used to define the presence or absence of a word in the sentence and therefore provide more intuition to the context. Syntactic information about words as well as information about the POS of neighbouring words, namely POS bi-grams, can also be employed to increase results.

For this dissertation, three major groups of features were used: Label Feature, Neighbouring Words and Key Words. None of these features depend on external linguistic resources. The idea behind this implementation was to

extract all the information necessary for classification direct from the raw data without using any additional information.

To extract the features, a Python program was coded for each word. In the end ten words were disambiguated, each one by three different algorithms resulting in a set of 30 different programs. It is possible to combine code into only one program, passing parameters and functions. However, for program debugging and improvement of the features, the strategy of creating one program for each action proved to be more adequate to the task. For a given word, the same features were used for all the three algorithms, so that the algorithms could be compared fairly.

3.7.1. Label Feature

The label feature represents the first position in a concordance line. The sentences extracted from the NILC corpus available at Linguateca were retrieved from journalistic texts and some occurrences are indentified by the name of the newspaper section that they belong: Economy, Politics, Sports, etc. There is also a code identifying the specific location of the sentence in the newspaper, and therefore a patter can be inferred by this information.

The idea to use this feature was inspired by the work of Koeling, McCarthy and Carroll (2005 and 2007). These researchers claimed that for some domains, the simple presence of an indication of the domain of the text is enough for a classifier to assert the correct class of ambiguous words:

> "Well defined and concise domains seem to be very helpful. Apparently, both the medical and the politics domain fit that bill. A good indication is the fact that a list of most salient words for that domain covers a reasonable size of the words in the document." (Koeling, McCarthy and Carroll, 2007)

The authors define salient words as words that appear very frequently in a given domain taken into account a certain number of documents from that domain. They calculate what they called salient ratio by dividing the number of

times a word *w* appears in a domain *d*, by the number of times a word *w* in all texts (domain independent).

Since the corpus used for these experiments already had domain information, there was and the opportunity to verify how informative can domain information be. The feature was extracted as follows:

```
features["label"] = line[0]
```

Figure 3.2: Label feature.

The sample code shown in Figure 3.2 represents the attribution of a feature called “label” to the token found on the first position in the line. To illustrate the usefulness of this feature, the following example extracted directly from the corpus is provided:

> par=Dinheiro-94b-eco-1: A portaria autoriza o uso de **etiquetas** em URV, mas os comerciantes terão de colocar em lugares visíveis o valor da URV do dia para que o consumidor possa fazer a conta e saber quanto pagará. (par=Money-94b-eco1: The law authorizes the use of **labels** in URV, but the shops will have to place the daily currency of the URV in a visible place in order to allow the consumer to calculate beforehand how much he will pay.)

This sentence was extracted from the section *dinheiro*, a Portuguese word for *money*. This newspaper section contains texts from economy and finance. Therefore, in the case of the word *etiqueta,* which can be roughly translated as *lable* and as *etiquette*, it is more likely that a text from the finance and economy domain will use *etiqueta* as *lable* and not meaning *etiquette.*

In the results which will be shown in chapter 4, this feature proved to be an informative option for the disambiguation of some words.

3.7.2. Neighbouring Words

The neighbouring words are features that look at a certain window to the left and to the right of the index word, defined by the range parameter. As the example showed in Figure 3.3.

This feature gives good results when applied to previously processed data removing what is commonly described in the literature as stop words. The concept of stop words comes from information retrieval and is used in contrast to content words. Stop words are words that help the text to be coherent and harmonious, but their meaning in context is very limited.

There is no general agreement or systematic description in the literature about what should be considered a stop word, as it also depends on the application. However, some POS categories are usually considered as such: prepositions, numerals, articles, auxiliary verbs and conjunctions. Nouns and adjectives are usually in the class of content words.

For a better understanding of this feature, the following Python code is provided:

```
    for I in range(1,3):
        try:
            features[“w(-%d)” % i] = line[index-i]
        except:
            features[“w(-%d)” % i] = “<None>”
        try:
            features[“w(+%d)” % i] = line[index+i]
        except:
            features[“w(+%d)” % i] = “<None>”
    return features
```

Figure 3.3: Neighbouring words feature, sample code.

The sample code in figure 3.3, defines as features the words found in a window (range) of 1 to 3 of the target word. The program tries to search for tokens to the left, – 1 and to the right, +1.

3.7.3. Key Words

The key words feature gives good results for the algorithms. They were extracted after analysis of the data as well as the number of occurrences of the given word in each of the senses. A Boolean value, true or false, was attributed to the presence or absence of the given word in the sentence, as figure 3.4 shows:

```
if "URV" in line:
    features["URV"] = True
else:
    features["URV"] = False
if "real" in line:
    features["real"] = True
else:
   features["real"] = False
```

Figure 3.4: Key words feature, sample code.

There was no standard number of key words used as features for each word. In each case, words were added or subtracted to help improve the results. In the example shown in figure 3.4, two words are used as key words, *URV* and *real* and the true or false value is attributed to the presence or absence of these words in each instance.

After presenting the features, the next chapter will be concerned with the discussion of results from the different experiments.

CHAPTER 4 - Results

This chapter presents the results obtained in the different steps of the task. For some steps, results were commented and evaluated.

Section 4.1, presents the frequency of the preliminary list of 13 words in the corpus. Section 4.2, contains the results for the Most Frequent Sense (MFS) baseline. Section 4.3, presents the results obtained by calculating the Kappa coefficient for inter-annotator agreement.

In section 4.4, the metrics used to evaluate the automatic disambiguation task are presented, namely accuracy, precision, recall and f-measure, along with a brief discussion on the role of these metrics in classification tasks.

In section 4.5 and its subsections, the results of the different algorithms applied are presented, first from a general and wider perspective and them narrowing to each of the individual cases in ten subsections, word by word.

4.1. Corpus Frequency

The first results to be presented are those resulted from corpus frequency. The 13 words selected appear along with its respective count and also the calculated frequency of occurrence in the corpus. Calculated based on the total number of tokes in the NILC corpus, which is approximately 43 thousand words.

Word	Count	Freq. %
Apêndice	85	0.000003
Arquivo	928	0.000029
Comissão	4209	0.000130
Crédito	4360	0.000134
Cultura	5670	0.000174
Essência	521	0.000016
Etiqueta	311	0.000010
Extrato	624	0.000019
Foco	704	0.000022
Garantia	1925	0.000059
Geração	2618	0.000081
Imagem	9114	0.000280
Painel	917	0.000028
Recurso	10644	0.000328
Regime	2736	0.000084
Volume	4198	0.000129
Overall	78909	0.002428

Table 4.1: Absolute and Relative Frequency of words in the corpus.

After this step the word *apêndice* was disregarded because it occured only 85 occurrences. A minimum amount of 100 examples was established in order to have enough training and testing examples for the classifier. In this case, even though *apêndice* and its English equivalent *appendix* is a classic example for disambiguation and a frequent word in everyday vocabulary, the NILC corpus containing journalistic texts does not contain enough occurrences of the word.

4.2. MFS Baseline

For the remaining 12 words, the Most Frequent Sense (MFS) Baseline was calculated using the methods described in section 3.3. It represents the baseline accuracy that the system can achieve without any disambiguation method.

Word	MFS
Arquivo	0.69
Comissão	0.97
Crédito	0.80
Cultura	0.86
Essência	0.78
Etiqueta	0.82
Extrato	0.71
Foco	0.69
Garantia	0.81
Geração	0.77
Imagem	0.69
Painel	0.73
Recurso	0.80
Regime	1.00
Volume	0.68

Table 4.2: MFS Basline results

For these 12 words, results varied according to each word in a range from 0.68 to 1.00. Two of the words presented a very high baseline result, *regime* and *comissão*. They were disregarded for the final experiments, mainly because of the lack of examples to constitute the minority class. Another reason is that for these cases, when assigning only the most frequent sense the system will already have a very high accuracy, recall and precision, which renders disambiguation virtually useless.

Ten words remained for the next and final steps, the Kappa Coefficient for Inter-Annotator Agreement and the disambiguation task itself. Results will be presented in next forthcoming sections.

4.3. KAPPA Coefficient for Inter-Annotator Agreement

The Kappa coefficient was calculated for the ten remaining words. It corroborates with the description presented in the 3rd chapter, showing that word sense disambiguation is not a trivial task regarding annotator agreement.

Word	Kappa
Arquivo	0.627
Crédito	0.731
Cultura	0.896
Essência	0.836
Etiqueta	0.776
Foco	0.776
Garantia	0.657
Geração	0.821
Imagem	0.552
Volume	0.493

Table 4.3: Kappa Coefficient results for Inter-Annotator Agreement.

Results varied from 0.493 to 0.896 and as described in section 3.4, eight words presented substantial agreement between annotators: *arquivo, crédito, cultura, essência, etiqueta, foco, garantia* and *geração*. The two lowest results *imagem* and *volume* are considered moderate agreement.

On the next sections of this chapter the results from the disambiguation methods will be presented as well as analysis on the results obtained.

4.4. Disambiguation Results - Metrics

Accuracy is the easiest and most common way of reporting the performance of machine learning methods. However, for some classification tasks, especially those involving highly imbalanced data, more precise metrics should be adopted in order to evaluate results more clearly.

When classifying skewed and highly imbalanced data, accuracy is usually very high and it does not reflect exactly the performance of the classifier. In these cases, evaluation should be concerned with the minority

class and assign it as a positive class, and all other classes as a negative class. Liu (2007) used the example of network intrusion detection to illustrate this fact:

> "Accuracy is not a suitable measure in such cases because we may achieve a very high accuracy, but may not identify a single intrusion. For instance, 99% of the cases are normal in an intrusion detection data set. Then a classifier can achieve 99% accuracy without doing anything by simply classifying every test case as "not intrusion" This is, however, useless." (Liu, 2007)

For these reasons, precision and recall have an important role in the evaluation of classifiers, because they can measure how precise and how complete the classification is on the positive class. To understand the formulae of precision and recall, the confusion matrix is used:

	Classified Positive	**Classified Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 4.4: Confusion Matrix

The values TP, FN, FP and TN presented in the confusion matrix can be interpreted as follows:

- True Positive (TP): Number of correct classifications of the positive examples.
- False Negative (FN): Number of incorrect classifications of positive examples.
- False Positives (FP): Number of incorrect classifications of negative examples.
- True Negative (TN): Number of correct classifications of negative examples.

Based on the values obtained on the confusion matrix, precision (P) and recall (R) for the positive class can be calculated as follows:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

In other words, precision can be defined as the number of correctly identified cases divided by the total number of identified cases. Recall is defined by the number of correctly identified cases divided by the number of cases which should have been identified. Results for precision and recall are in theory not related, which means that a classifier can obtain high results in precision and low results in recall as well as low precision and high recall.

In WSD, precision represents the correct classified cases among all the classified examples. In a practical example, precision will be 1.0 for a class X, if all the instances classified as class X belong to this class and gives no information about instances which should have been classified as X, but were classified with other label.

Recall represents the relation between the correct classified instances and all the instances that should have been identified in a given class. In WSD, Recall will be 1.0 for a class X, if all the instances of class X were correctly identified as belonging to this class. However, recall provides no information about instances that were labeled as class X and in fact belong to other classes.

The importance of each measure depends on the nature of the application, and usually the f-measure, also called the f-score, is used as a single measure to compare classifiers. The formula used to calculate the f-measure as follows:

$$F = \frac{(\beta + 1) \times P \times R}{(\beta \times R) + R}$$

The relative importance of each of the components can be altered by varying the value of the β variable. If β is set to 1 then recall and precision count equally, at β = 0.5 precision is twice as important as recall and at β = 2 recall is

twice as important as precision. The value for β is commonly set to 1, and that is the value that will be considered for evaluation in this work. Resulting in the following simplified formula:

$$F = \frac{2PR}{P+R}$$

In the next section the results and the performance of the classifiers will be presented and discussed in terms of average accuracy and also precision, recall and f-measure.

4.5. General Observations

Presentation and comments on the results will begin with general observations and then the further ten subsections, one for each word, will concentrate on the particulars of each case. For an overview of the best classification results, table 4.5 presents the best accuracy obtained for each of the ten words along with the respective algorithms. The MFS baseline is presented once again to provide the baseline for the classifiers.

Word	MFS	Best Accuracy	Algorithm
Arquivo	0.69	0.85	NB
Crédito	0.80	0.96	Maxent
Cultura	0.86	0.99	Maxent
Essência	0.78	0.93	Maxent
Etiqueta	0.82	0.94	DT
Foco	0.69	0.65	NB and Maxent
Garantia	0.81	0.97	Maxent
Geração	0.77	0.93	Maxent
Imagem	0.69	0.72	DT
Volume	0.68	0.81	NB

Table 4.5: Best accuracy results and algorithms.

At first glance, the best methods performed above the baseline in terms of accuracy for all the cases, except the word *foco*, for which the results were four per cent below the baseline. As explained in the previous section, accuracy is not the only measure that needs to be taken into account when analyzing the performance of classifiers, especially in unbalanced data such as the one used for these experiments. Therefore, a more detailed analysis will be carried out word by word in the following subsections. Before discussing in more detail, some other information can be obtained from a global perspective.

Regarding the algorithms, Maximum Entropy performed better in six out of the ten cases and was considered the best classifier for this task. Naïve Bayes was the best classifier for three words, however its performance for most of the words is very close to the best results that were obtained using Maxent.

Decision Tree was the best method for two words, *etiqueta* and *imagem*, however its overall performance for the ten words is significantly below the other two algorithms. Therefore among the algorithms used, Decision Tree proved to be the least suitable for the task. And more on that will be commented based on the table 4.6 presented below:

Word	MFS	Naive Bayes	Maxent	Decision Tree
Arquivo	0.69	0.85	0.79	0.68
Crédito	0.80	0.91	0.96	0.45
Cultura	0.86	0.96	0.99	0.65
Essência	0.78	0.78	0.93	0.85
Etiqueta	0.82	0.93	0.92	0.94
Foco	0.69	0.65	0.65	0.61
Garantia	0.81	0.96	0.97	0.65
Geração	0.77	0.73	0.93	0.69
Imagem	0.69	0.67	0.69	0.72
Volume	0.68	0.81	0.80	0.73
Average	**0.76**	**0.82**	**0.86**	**0.70**

Table 4.6: Accuracy results for all algorithms.

When calculating the average of the accuracy scores of the three methods, Naïve Bayes and Maximum Entropy are substantially better than the MFS baseline, six per cent higher in the case of Naïve Bayes and ten per cent for Maxent. The Decision Tree algorithm was on average six per cent below the baseline, which means that when applying decision three for the complete set of words, the average accuracy would be worse than using no disambiguation method at all, by simple assigning the majority class as the correct word sense.

Besides its poor performance, the Decision Tree (in its implementation in Python available on NLTK) presented another disadvantage compared to the other two models use. It does not display the most informative features taken into account in the classification. As a result, it is difficult to refine the features to be used in the classification, unless the output is formatted as a decision tree and analyzed. Given the significantly large number of variables that the algorithms have to deal with in WSD, analyses of a complete decision tree would be a very time consuming activity.

A word by word result description will be presented in the following sub-sections.

4.5.1. Arquivo

Given an MFS baseline of 69 per cent for the word *arquivo*, Naïve Bayes classifier and Maximum Entropy performed above the baseline results in its average accuracy. The Decision Tree was below the baseline for this word, however not far bellow (as some other cases). As the following three tables show:

ARQUIVO	Majority Class	Minority Class
Precision	0.83	0.72
Recall	0.94	0.56
F-Measure	0.88	0.61
Average Accuracy	**0.85**	

Table 4.7: Naïve Bayes Results for the word *arquivo*.

.ARQUIVO	Majority Class	Minority Class
Precision	0.81	0.77
Recall	0.87	0.43
F-Measure	0.84	0.53
Average Accuracy	**0.79**	

Table 4.8: Maximum Entropy results for the word *arquivo.*

ARQUIVO	Majority Class	Minority Class
Precision	0.70	0.52
Recall	0.90	0.27
F-Measure	0.78	0.35
Average Accuracy	**0.68**	

Table 4.9: Decision Tree results for the word *arquivo.*

When comparing the MFS baseline against the results for the minority class, Maxent and Naïve Bayes are precise enough 0.72 and 0.77. However its coverage is limited to 0.56 and 0.43, which resulted in an F-Measure of 0.61 and 0.53 respectively, both bellow the baseline. Recall was also considerably lower than precision in Decision Tree, which shows that in order to obtain better results in disambiguation for this word, recall must be significantly increased.

4.5.2. Crédito

Unlike the previous example of the word *arquivo*, the best classifier for the word *crédito* was in fact Maximum Entropy, reaching 0.96 of average accuracy. Naïve Bayes also performed above the baseline, with 0.85 of average accuracy against an MFS baseline of 0.80, as shown on the following three tables:

CRÉDITO	Majority Class	Minority Class
Precision	0.93	0.86
Recall	0.92	0.90
F-Measure	0.92	0.85
Average Accuracy	**0.91**	

Table 4.10: Naïve Bayes results for the word *crédito*.

CRÉDITO	Majority Class	Minority Class
Precision	0.95	1.00
Recall	0.99	0.89
F-Measure	0.97	0.93
Average Accuracy	**0.96**	

Table 4.11: Maximum Entropy results for the word *crédito*.

CRÉDITO	Majority Class	Minority Class
Precision	0.27	1.00
Recall	1.00	0.05
F-Measure	0.42	0.09
Average Accuracy	**0.45**	

Table 4.12: Decision Tree results for the word *crédito.*

Unlike the previous example of *arquivo* which presented results above the baseline in terms of accuracy and bellow the baselines in terms of F-Measure, the word *crédito* reached satisfactory results above baseline both in accuracy and in F-Measure. In terms of F-Score for the minority class, the two best methods, Naïve Bayes and Maximum Entropy, presented satisfactory results above the 0.80 baseline, reaching 0.85 and 0.93.

Decision Tree was not a good method for the word *crédito.* Precision for minority class and recall for majority class are reported as 1.00. However, the complementary measures are very poor, 0.27 and only 0.05, resulting in very low F-scores.

Decision tree results presented this very same interesting pattern for some of the other words, a high and sometimes very high precision for the minority class with low recall, and a very high recall for the majority class with very low precision. One hypothesis is that when dealing with small datasets, the minority class does not have information for building decision trees to classify all cases correctly, however the features present specific information that can identify precisely a few number of cases.

4.5.3. Cultura

The word *cultura* obtained the best results in these experiments, as it already begun with the highest baseline among the 10 words, 0.86. A high baseline means that if the classifier assigns the majority class to all the cases, it should have at least 0.86 of accuracy, but on the other hand it means that the data for disambiguation is even more unbalanced than lower baselines. This reason led to the exclusion of two words prior to the classification experiments as explained in section 4.2.

The word *cultura* also had the best results in inter-annotator agreement, 0.89. Therefore it gives evidence to the hypothesis that the more annotators agree on senses of an ambiguous word in context, the more likely that these senses possess clear distinguishable features that will help machine learning algorithms to perform automatic disambiguation.

Results for the word *cultura* are presented in the following tables:

CULTURA	Majority Class	Minority Class
Precision	0.96	0.98
Recall	0.99	1.00
F-Measure	0.97	0.99
Average Accuracy	**0.96**	

Table 4.13: Naïve Bayes results for the word cultura.

CULTURA	Majority Class	Minority Class
Precision	1.00	0.98
Recall	0.98	1.00
F-Measure	0.99	0.99
Average Accuracy	**0.99**	

Table 4.14: Maximum Entropy results for the word *cultura*.

CULTURA	Majority Class	Minority Class
Precision	0.63	0.82
Recall	0.67	0.97
F-Measure	0.65	0.86
Average Accuracy	**0.65**	

Table 4.15: Decision Tree results for the word *cultura.*

Good results were obtained both in terms of accuracy and F-Measure for both classes when using Naïve Bayes and Maxent classifiers. Decision trees

once again, presented significantly lower results than the other two algorithms, providing another evidence of its inadequacy for this task.

4.5.4. Essência

The disambiguation task for the word *essência* was performed better by Maximum Entropy, as shown in table 4.15. Precision, recall and f-measure, obtained using Maximum Entropy, were also better than the 0.78 MFS baseline result for this word.

ESSÊNCIA	Majority Class	Minority Class
Precision	0.96	0.99
Recall	0.40	0.81
F-Measure	0.56	0.88
Average Accuracy	**0.78**	

Table 4.16: Naïve Bayes results for the word *essência*.

ESSÊNCIA	Majority Class	Minority Class
Precision	0.97	0.97
Recall	0.96	0.84
F-Measure	0.97	0.89
Average Accuracy	**0.93**	

Table 4.17: Maximum Entropy results for the word *essência*.

ESSÊNCIA	Majority Class	Minority Class
Precision	0.95	0.72
Recall	0.46	0.95
F-Measure	0.52	0.80
Average Accuracy	**0.84**	

Table 4.18: Decision Tree results for the word *essência.*

4.5.5. Etiqueta

From the ten words studied in this work, *etiqueta* was the word in which the three methods presented the most similar results between then. Average accuracy results varied in only 2 per cent and Decision Tree performed slightly better than Naïve Bayes and Maxent. However, when taking into account the f-

measure results for the minority class, Maximum Entropy was once again better. The result for inter-annotator agreement for the word *etiqueta* was 0.776, considered substantial agreement among annotators.

ETIQUETA	Majority Class	Minority Class
Precision	1.00	0.80
Recall	0.81	1.00
F-Measure	0.89	0.89
Average Accuracy	**0.93**	

Table 4.19: Naïve Bayes Results for the word *etiqueta.*

ETIQUETA	Majority Class	Minority Class
Precision	0.95	0.96
Recall	0.88	1.00
F-Measure	0.91	0.98
Average Accuracy	**0.92**	

Table 4.20: Maximum Entropy results for the word *etiqueta.*

ETIQUETA	Majority Class	Minority Class
Precision	0.95	0.94
Recall	0.93	1.00
F-Measure	0.93	0.97
Average Accuracy	**0.94**	

Table 4.21: Decision Tree results for the word *etiqueta.*

4.5.6. Foco

The word *foco* was the only word in which the disambiguation methods performed below the baseline results. MFS baseline for *foco* was 0.69 and the three methods presented lower results in terms of accuracy. Like the previous word *etiqueta*, methods presented close results for accuracy:

FOCO	Majority Class	Minority Class
Precision	0.73	0.90
Recall	0.75	0.45
F-Measure	0.77	0.60
Average Accuracy	**0.65**	

Table 4.22: Naïve Bayes Results for the word *foco.*

FOCO	Majority Class	Minority Class
Precision	0.86	0.69
Recall	0.76	0.47
F-Measure	0.81	0.54
Average Accuracy	**0.65**	

Table 4.23: Maximum Entropy results for the word *foco.*

FOCO	Majority Class	Minority Class
Precision	0.76	0.59
Recall	0.71	0.36
F-Measure	0.73	0.42
Average Accuracy	**0.61**	

Table 4.24: Decision Tree results for the word *foco.*

For this case an error analysis stage would be necessary in order to quantify and qualify the misclassified instances and replicate the experiments for a second run.

4.5.7. Garantia

The word *garantia* presented very similar results for the algorithms Naïve Bayes and Maximum Entropy. In both classes, precision, recall and f-measure presented similar results.

GARANTIA	Majority Class	Minority Class
Precision	0.99	0.94
Recall	0.93	1.00
F-Measure	0.96	0.97
Average Accuracy	**0.96**	

Table 4.25: Naïve Bayes results for the word *garantia.*

GARANTIA	Majority Class	Minority Class
Precision	1.00	0.96
Recall	0.94	1.00
F-Measure	0.97	0.98
Average Accuracy	**0.97**	

Table 4.26: Maximum Entropy results for the word *garantia.*

GARANTIA	Majority Class	Minority Class
Precision	1.00	0.47
Recall	0.24	1.00
F-Measure	0.39	0.63
Average Accuracy	**0.65**	

Table 4.27: Decision Tree results for the word *garantia.*

In table 4.27, once again it is possible to observe what was commented in section 4.5.2 for the word *credito* regarding the Decision Tree algorithm. High results for precision in the majority class and recall in the minority class, in this case 1.0. Along with low values for the complementary measures, recall for majority class and precision for minority class.

4.5.8. Geração

For the word *geração*, Maximum Entropy presented significantly higher scores compared the other two methods. As it is shown in the following three tables, Maxent was the only method which performed better than the 0.77 MFS baseline result in terms of accuracy.

GERAÇÃO	Majority Class	Minority Class
Precision	0.94	0.61
Recall	0.37	1.00
F-Measure	0.43	0.74
Average Accuracy	**0.73**	

Table 4.28: Naïve Bayes results for the word *geração.*

GERAÇÃO	Majority Class	Minority Class
Precision	0.85	0.98
Recall	0.95	0.83
F-Measure	0.90	0.90
Average Accuracy	**0.93**	

Table 4.29: Maximum Entropy results for the word *geração*

GERAÇÃO	Majority Class	Minority Class
Precision	1.00	0.84
Recall	0.09	0.96
F-Measure	0.15	0.87
Average Accuracy	**0.69**	

Table 4.30: Decision Tree results for the word *geração.*

Maxent also presented satisfactory results in terms of minority class´ f-measure, 0.90 and once again above the baseline.

4.5.9. Imagem

Among the ten words, the word *imagem* was the only case that the methods Decision Tree performed significantly better than the other two methods. This performance is verified not only in accuracy, but more significantly in terms of minority class´ f-measure, obtaining 0.65 against 0.41 and 0.48.

IMAGEM	Majority Class	Minority Class
Precision	0.60	0.71
Recall	0.85	0.30
F-Measure	0.70	0.41
Average Accuracy	**0.67**	

Table 4.31: Naïve Bayes Results for the word *imagem*.

IMAGEM	Majority Class	Minority Class
Precision	0.67	0.78
Recall	0.84	0.36
F-Measure	0.75	0.48
Average Accuracy	**0.69**	

Table 4.32: Maximum Entropy results for the word *imagem*.

IMAGEM	Majority Class	Minority Class
Precision	0.70	0.74
Recall	0.69	0.65
F-Measure	0.66	0.65
Average Accuracy	**0.72**	

Table 4.33: Decision Tree results for the word *imagem.*

For the word *imagem*, the kappa coefficient was the second worst, after *volume*. This shows evidence that kappa coefficient results can provide an insight on the performance of disambiguation methods, the same observation, but on the opposite direction to what was discussed in section 4.5.3 for the word *cultura*. *Cultura* had the best scores for kappa and also for the disambiguation task itself.

4.5.10. Volume

For the word volume, the three best methods presented accuracy results above the 0.68 baseline. However, in terms of f-measure for the minority class, the results do not represent any improvement compared to the baseline. As it is shown in the following three tables:

VOLUME	Majority Class	Minority Class
Precision	0.86	0.56
Recall	0.77	0.66
F-Measure	0.81	0.59
Average Accuracy	**0.81**	

Table 4.34: Naïve Bayes Results for the word *volume.*

VOLUME	Majority Class	Minority Class
Precision	0.74	0.59
Recall	0.94	0.20
F-Measure	0.83	0.29
Average Accuracy	**0.80**	

Table 4.35: Maximum Entropy results for the word *volume.*

VOLUME	Majority Class	Minority Class
Precision	0.80	0.80
Recall	0.56	0.69
F-Measure	0.53	0.60
Average Accuracy	**0.73**	

Table 4.36: Decision Tree results for the word *volume.*

The word *volume* presented the worst result among the ten words in kappa inter-annotator agreement, 0.493. Along with a low performance in minority class´ f-measure, the lowest kappa results corroborate to the previously

mentioned relation between annotator-agreement and performance of the disambiguation methods.

5. Conclusion

This dissertation aimed at exploring different machine learning algorithms and techniques to the automatic disambiguation of a set of Portuguese nouns. It began by the analysis of an academic vocabulary in order to select ambiguous words from the vocabulary and thirteen words were selected. After calculating the MFS baseline, ten words remained for the experiments, which include Kappa Inter-annotator agreement and the use of three algorithms to perform the automatic disambiguation of these nouns.

The experiments carried out in this dissertation have shown satisfactory results according to literature on the state-of-the-art in WSD. The results were achieved through a labor-intensive process of refining the input features incorporating a linguistic analysis of the corpus performed by native Portuguese speakers.

The results presented here constitute an encouraging perspective for other machine learning approaches to WSD as well as other tasks in NLP. This is mainly because the corpus data used for training and testing is untagged (POS tags, syntactic and semantic parsers were not used). Secondly, it is encouraging because the amount of data collected for the experiments was not significantly large as most other applications using machine learning, which proves that it is possible to perform automatic disambiguation using medium sized corpora. This can be particularly useful for resource-poor languages that have fewer linguistic resources available than Portuguese.

Some other conclusions can be drawn from this dissertation and they can be applied in further WSD and NLP research. The first is that the assumption that domain information is an important feature to perform automatic disambiguation is true for the set of words and corpora used in this work. Therefore, it corroborates the conclusions of Koeling, McCarthy, and Carroll (2007).

Another important point is that the results obtained by the Kappa coefficient on inter-annotator agreement, seems to indicate how well classifiers will perform disambiguation for a given word. When the agreement on word

senses is high, it is more likely that the senses will have strong distinctive features that will provide evidence for the algorithms to disambiguate it.

Regarding the algorithms, Python NLTK implementations of Naïve Bayes and Maximum Entropy classifiers reached good levels of performance in WSD with the features used and described here. Results obtained using Decision Tree, were not as satisfactory as the other two methods and should be analyzed more carefully.

Besides the experiments, methods and conclusions described in this dissertation, another important contribution from this work can be mentioned. We took part in the compilation of the Portuguese Academic Wordlist (PAWL) (Baptista, 2010), developing a linguistic resource for the Portuguese NLP and Linguistics and research community

An aspect that could complement this work in order to obtain better results is error analysis. With the time dedicated to perform all the experiments and methodological steps described throughout this dissertation, it was not possible to establish a methodology to perform a careful examination of misclassified examples in order to quantify and qualify errors. This analysis could improve the methods and therefore help to generate better results.

Finally, the experiments described here were performed using open source applications, namely Python and NLTK, which can diminish software costs for research.

5.1. Further perspectives

The results presented here encourage further research for a larger number of words. The first application that can benefit from this work is naturally the REAP.PT. Provided that the words here analyzed and disambiguated are a subset of the words to be used in REAP.PT, based on the Portuguese Academic Wordlist, P-AWL, it is possible to replicate this process for other words and thus cover a wider range of the lexicon.

This work was the first step towards the design of a module for REAP.PT. Further experiments should be carried out to show the importance of WSD within the framework of CALL software, not only from a technological perspective, but also from a pedagogical point of view, as in the experiments described by Kulkarni, et. al. (2008).

Other NLP applications, besides REAP.PT, in which ambiguity constitutes an obstacle for satisfactory performance, can replicate the methods described in this dissertation, in Portuguese or other languages.

REFERENCES

Agirre, E.; Edmonds, P.; (eds.). (2006) Word Sense Disambiguation: Algorithms and Applications. Springer.

Aone, C.; Bennet, S. (2000) Applying machine learning to anaphora resolution, Lecture Notes in Computer Science, Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing, 1040/1996. Pages 302 – 314

Banerjee, S.; Pedersen, T. (2002). "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. Lecture Notes In Computer Science; Vol. 2276.

Baptista, J.;Costa, N.; Guerra, J.; Zampieri, M.; Cabral, M.; Mamede, N. (2010) in T.A.S. Pardo et. al (Editors) "P-AWL: Academic Word List for Portuguese", PROPOR2010, LNAI 6001, p. 120-123.

Berger, A.; Pietra, S.; Pietra, V. (1996). A maximum entropy approach to natural language processing. Computational Linguistics, 22, 39–71.

Bird, S; Klein, E; Loper, E. (2009) Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit, O´Reilly Media.

Brill, E. (1993) Automatic grammar induction and parsing free text: a transformation-based approach. Proceedings of the 31st Annyal Meeting of the Association for Computational Linguistics (ACL 93) (Columbus, OH), 259 - 265.

Carletta, J. (1996) Assessing agreement on classification tasks - The kappa statistic, Computational Linguistics, 22(2) 249-254

Collins-Thompson, K.; Callan, J. (2004) Information retrieval for language tutoring: An overview of the REAP project (poster description). Proceedings of the Twenty Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Sheffield, UK.

Coxhead, A. (2000) A New Academic Word List, TESOL Quarterly, 34, p. 213-238.

Duda, R.; Hart, P. (1973) Pattern Classification and Scene Analysis. Wiley.

Jurafsky, D.; Martin, J. (2009) Speech and Language Processing (2nd Edition) Pearson Education

Koeling, R.; McCarthy, D.; Carroll, J. (2005) Domain-specific sense distributions and predominant sense acquisition. In: Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing., Vancouver, Canada . Pages 419 - 426

Koeling, R.; McCarthy, D.; Carroll, J. (2007) Text categorization for improved priors of word meaning. In Proceedings of the Eighth International Conference on Intelligent Text Processing and Computational Linguistics (Cicling 2007), pages 241.252, Mexico City, Mexico.

Kulkarni, A.; Heilman, M.; Eskenazi, M.; Callan, J. (2008) Word Sense Disambiguation for Vocabulary Learning. Ninth International Conference on Intelligent Tutoring Systems.

Finlay, J.; Dix, A. (1996) An Introduction to Artificial Intelligence, UCL Press, pages 90 - 95.

Florian, R.; Cucerzan, S.; Schafer, C.; Yarowsky, D. (2002). Combining classifiers for word sense disambiguation. Journal of Natural Language Engineering, 8(4):327–342.

Freitag, D. (1998) Toward general-purpose learning for information extraction. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and COLING-98 (ACL/COLING´ 98) Montreal, pages 404 - 408.

Geoffrey, L.; Weisser, M. (2003) Pragmatics and Dialogue in Mitkov (Editor) Oxford Handbook of Computational Linguistics, Oxford University Press, pages 137 - 156.

Gomez Hidalgo, J.; Buenaga, M.; Cortizo, J.C. (2005) The role of word sense disambiguation in automated text categorization. In Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems, Lecture Notes in Computer Science, Springer, pages 298 - 309.

Hirst, G. (1987) Semantic Interpretation and the Resolution of Ambiguity. Cambridge University Press.

Kearns, K. (2000) Semantics, Modern Linguistics Series, MacMillian, London - England.

Lesk, M. (1986) Automatic sense disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In: Proceedings of ACM SIGDOC Conference, p. 25-26. Toronto, Canada

Liu, B. (2007) Web Data Mining: Exploring Hyperlinks, Contents and Usage Data, Springer

Magerman, D. (1995) Statistical decistion-tree models for parsing. Proceedings of the 33rd Annual Meering of the Association for Computational Linguistics (ACL 95) (Cambridge, Mass.), 276 – 283.

Magnini, B.; Strapparava, C.; Pezzulo, G.; Gliozzo, A. (2002) The role of domain information in word sense disambiguation. Natural Language Engineering 8, 359 – 373.

McCallum, A. (2002) MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.

Mamede, N.; Baptista, J.; Vaz, P.; Hagège, C. (2010) Nomenclature of chunks and dependencies in Portuguese XIP grammar (v. 2.1.). Internal Report. Lisboa: L2F/INESD-ID Lisboa.

Manning, C.; Schütze, H.; (1999) Foundations of Statistical Natural Language Processing, MIT Press.

Marcus, M.; Santorini, B.; Marcinkiewicz, M. (1993) Building a large annotated corpus of English: the Penn treebank. Computational Linguistics, 19(2), 313 – 330.

Màrquez, L.; Padró, L.; Rodríguez, H.; (2000) A Machine Learning Approach to POS Tagging, Machine Learning, v.39 n.1, p.59-91.

Marujo, L. (2009) REAP.PT – REAP.PT, Master Thesis, Instituto Superior Técnico (IST), Lisboa.

Marquez, L.; Padro, L.; Rodriguez, H. (1999) A machine learning approach to POS tagging. Machine Learning, 39(1), 59 – 91.

Miller, G.; Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, K. (1993) Introduction to WordNet: an Online Lexical Database, International Journal of Lexicography, 234 – 244.

Mitkov, M. (2003) Anaphora Resolution in Mitkov (Editor) Oxford Handbook of Computational Linguistics, Oxford University Press, pages 249-265.

Mooney, R. (2003) Machine Learning in Mitkov (Editor) Oxford Handbook of Computational Linguistics, Oxford University Press, pp.376-394.

Ng, H.; Lee, H. (1996) Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Examplar-Based Approach. In Procceedings of the 24th

Ng, H.; Lim, C.; Foo, S. K. (1999) A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation. *Proceedings of the ACL SIGLEX*

Workshop: Standardizing Lexical Resources, College Park, MD, USA, p. 9-13.

Ng, H.; Zelle, J. (1997) Corpus-based approaches to semantic interpretation in natural language processing. AI Magazine, 18(4), 45 – 64.

Nigam, K.; Lafferty, J.; McCallum, A. (1999) Using maximum entropy for text classification. *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61-67.

Procter, P. (Ed.) (1978) Longman Dictionary of Contemporary English. London, Longman Group.

Pustejovski, J. (1995) The Generative Lexicon, MIT Press, London - England.

Pustejovski, J.; Boguraev, B.; (1996) Lexical Semantics – The Problem of Polisemy, Oxford University Press.

Quinlan, J. (1986) Induction of Decision Trees. Machine Learning 1, 1 (Mar. 1986). Pages 81-106.

Quinlan, J. (1992). C4.5 Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann.

Ratnaparkhi, A. (1996) A maximum entropy model for part-of-speech tagging. In *Proceedings of the Empirical Methods in Natural Language Conference*.

Santos, D. (2000) "O projecto Processamento Computacional do Português: Balanço e perspectivas", in Volpe Nunes (ed.), *Actas do V Encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR'2000)* (Atibaia, São Paulo, Brasil, 19 a 22 de Novembro de 2000), pp. 105-113.

Santos, D.; Simões, A.; Frankenberg-Garcia, A.; Pinto, A.; Barreiro, A.; Maia, B.; Mota, C.; Oliveira, D.; Bick, E.; Ranchhod, E.; Almeida, J.;

Cabral, L.; Costa, L.; Sarmento, L.; Chaves, M.; Cardoso, N.; Rocha, P.; Aires, R.; Silva, R.; Vilela, R.; Afonso, S. (2004) "Linguateca: um centro de recursos distribuído para o processamento computacional da língua portuguesa". In Guillermo De Ita Luna, Olac Fuentes Chávez & Mauricio Osorio Galindo (eds.), *Proceedings of the international workshop "Taller de Herramientas y Recursos Linguísticos para el Espanõl y el Portugués", IX Iberoamerican Conference on Artificial Intelligence (IBERAMIA 2004)* (Puebla, México, Novembro de 2004), pp. 147-154

Specia, L. (2007) Uma abordagem híbdrida relacional para a desambiguação lexical de sentido na tradução automática; PhD Thesis

Stevenson, M.; Wilks, Y. (2003) Word Sense Disambiguation in Mitkov (Editor) Oxford Handbook of Computational Linguistics, Oxford University Press, pages 249-265.

Witten, I.; Frank, **E.** (2005) Data Mining: Practical machine learning tools and techniques, (2nd Edition), Morgan Kaufmann, San Francisco

Yarowsky, D. (1996a) Three Algorithms for Lexical Ambiguity Resolution, PhD. Thesis, School of Computer and Information Science, University of Pennsylvania

Yarowsky, D. (1996b) Homograph Disambiguation in Speech Synthesis. In J. van Santen, R. Sproat, J. Olive and J. Hirschberg (eds.), Progress in Speech Synthesis. Springer-Verlag, pp. 159-175.

Yarowsky, D. (1997). Homograph disambiguation in text-to-speech synthesis. In Jan T. H. van Santen, Richard Sproat, Joseph P. Olive, and Julia Hirschberg (Editors) Progress in Speech Synthesis. Springer-Verlag, New York, pp.157-172.

Zhang, H. (2004) The optimality of naive Bayes, Proceedings of the 17th International FLAIRS conference (FLAIRS2004), American Association for Artificial Inteligence AAAI Press.

Zipf, G. (1949) Human Behavior and the Principle of Least-Effort. Addison-Wesley.

ANNEX 1 - Corpus Occurrences of Arquivo

par=114022: A viúva começou recusando; mas o padre instou, expôs o que era, disse-lhe que nada perdia o devido respeito à memória do marido consentindo que alguém folheasse uma parte da biblioteca e do arquivo, uma parte apenas; e afinal conseguiu, depois de longa resistência, que me apresentasse lá .

par=100048: III -- o respectivo material, inclusive máquinas e equipamentos, arquivos, documentos e processos, instalações e demais bens afetados aos referidos órgãos ;

par=52912: Em 1935, Elisabeth, meses antes de morrer, foi homenageada com uma visita do Führer aos seus Arquivos; e Hitler fez questão de comparecer ao seu enterrro para significar a estima que sentia pela irmão do ilustre filósofo .

par=49669: Por lógico não se professa mais o absoluto positivista que delegava ao documento antigo, original, guardado em arquivo e escrito o foro de verdade .

>: <STRONG< i>Arquivos policiais de Brasília escondem crimes sexuais praticados por socialites e subalternos burocráticos

>: Por cinco anos, os partidos e os candidatos deverão manter <STRONG< i>arquivos com as suas prestações de contas e a relação completa de todas as doações recebidas e a identificação dos doadores

par=Ilustrada-94b-nd-1: Só nos arquivos de Paulo Tapajós encontraram, poucas semanas atrás, as seguintes :

par=Brasil-94a-pol-1: Para evitar isto, funcionários serão responsabilizados pela preservação dos arquivos .

par=Ilustrada-94b-nd-1: Apesar da extensão de seu arquivo, Mário Luiz deixa claro que seleciona os músicos retratados .

par=Ilustrada-94b-nd-1: O Estado do Tennessee mandou rever os arquivos médicos de Elvis Presley para determinar se ele morreu de overdose de medicamentos, em 77 .

par=Brasil-94a-pol-1: PC Farias deixou registrado no seu depoimento à CPI, como os jornais transcreveram na época e hoje têm nos seus arquivos, a confissão de que o esquema dos fantasmas deu dinheiro para a campanha do então candidato pelo PFL, e hoje governador de Pernambuco, Joaquim Francisco .

par=48864: Em primeiro lugar, eles ampliaram o arquivo de evidências empíricas que favorecem a existência e a flexibilidade da imagética mental .

par=Ilustrada-94b-nd-2: Os responsáveis pelo programa vasculharam uma multidão de arquivos de filmes (russos, norte-americano, franceses, ingleses, alemães) e selecionaram imagens produzidas pelas agências oficiais desses países que compõem uma formidável ilustração da história do século 20, tal como vista por seus protagonistas .

par=29295: Lá, sem livros nem arquivos, escreveu a obra que lançou seu nome como um dos fundadores da Nova História: O Mediterrâneo e o mundo mediterrâneo na época de Felipe II, publicado em 1949 .

par=Cotidiano-94b-soc-1: Além dessa área, o Oficina espera que a prefeitura ceda três terrenos um embaixo do viaduto, onde hoje há um sacolão, e dois do outro lado da rua, retalhos de desapropriações para oficinas, salas de ensaio, bar e arquivo .

par=Esporte-94b-des-2: Reviro meus arquivos .

par=Brasil-94b-pol-2: Dallari Tenho um arquivo pessoal .

par=Cotidiano-94b-soc-2: O material que não for usado na estréia do quadro hoje vai para um arquivo .

par=60871: Pensamos em pegar, em nossos arquivos, matérias antigas sobre o assunto que pudessem ser readaptadas .

par=Ilustrada-94a-nd-1: Os comunistas lhe abriram todos os arquivos sobre os colonos franceses na Indochina .

par=Especial-94a-nd-2: Eu insistia para que o governo do Estado recebesse de volta os arquivos .

par=Cotidiano-94a-soc-1: A Fundação Getúlio Vargas abre no dia 21, segunda-feira, uma exposição fotográfica com cerca de cem fotos do arquivo da Escola de Administração de Empresas de São Paulo .

par=Ilustrada-94b-nd-2: Na hora de gravar o disco, fui no arquivo da coroa e peguei algumas coisas. "

par=Ilustrada-94b-nd-1: Autoridades nessa área me garantiram que o arquivo corre perigo se sair das minhas mãos .

par=44383: O segundo conjunto de estruturas de dados consiste nos arquivos da memória de longo prazo, que representam a informação usada na geração de imagens .

par=42219: E parte à descoberta do Brasil concreto: o do passado, nos arquivos, o do presente, viajando pelo país .

par=Ilustrada-94a-nd-1: A prioridade é a restauração do prédio do antigo Matadouro Municipal, onde a Cinemateca está instalada desde o início de 94, a fim de facilitar ao público o acesso aos arquivos e ao acervo de imagens sob os cuidados da instituição .

par=Esporte-94b-des-1: E as imagens de Senna logo postas no ar sorrindo, vencendo começavam estranhamente a se transformar em arquivo .

par=Ilustrada-94b-nd-1: A segunda quantia mais elevada, US$ 181 mil, se destinaria à Fundação Memória Republicana, o Museu Sarney no Maranhão, onde o ex-presidente guarda seus arquivos e construiu seu mausoléu .

>: Por cinco anos, os partidos e os candidatos deverão manter <STRONG< i>arquivos com as suas prestações de contas e a relação completa de todas as doações recebidas e a identificação dos doadores

par=8110: Organizou recentemente um livro póstumo de José Honório, História diplomática do Brasil, com base no que encontrou no arquivo .

par=73917: 355: 2a. Lei 8.159, de 8.1.91 -- Dispõe sobre a política nacional de arquivos públicos e privados e dá outras providências (Lex 1991/12 e 106, ret .

par=Cotidiano-94b-soc-2: Todas essas imagens também poderão ser vistas por quem está sendo tratado o que, para Matson, vai aumentar a consciência do paciente sobre seu problema, além de permitir um arquivo mais preciso do histórico do tratamento .

par=Empregos-94b-eco-1: Apesar de a decisão de contratação ser tomada geralmente em duas semanas, a vida útil de um currículo no arquivo de uma empresa costuma ser de seis meses a um ano .

par=99023: Tal arquivo deverá incluir também todos os antecedentes relativos ao certificado emitido como também aqueles relativos à declaração exigida, de conformidade com o estabelecido no artigo anterior .

par=Fovest-94a-eco-1: Áreas de atuação: organização e conservação de arquivos de redações, assessoria de imprensa (divulgação de informações, ponte entre empresas e jornalistas) e trabalho na imprensa (reportagem, redação, edição, colunismo, revisão etc.)

par=38730: Entretanto a Quinta do Surdo, casa de campo nas paredes daA segunda versão nasceu a partir de pesquisas de Juan Pérez de Gusmán, que afirmou ter encontrado no arquivo da prefeitura de Madri um documento provando que, em 1808, Goya residia na Puerta del Sol .

par=Empregos-94a-eco-1: Dessa forma, os procedimentos em caso de acidente de trabalho não mudam mesmo que o profissional tenha escorregado no skate do filho ao buscar um documento no arquivo, em sua casa .

par=Cotidiano-94b-soc-1: A Comissão Pastoral da Terra em Belém vai convidar o delegado do Trabalho, Raimundo Gomes Filho, para conhecer os arquivos da entidade sobre denúncias de trabalho escravo .

par=9425: Será um livro de referência e consulta, pois vou incluir seu currículo, as premiações, a discografia e a musicografia, fornecidas por Vera Alencar, responsável pelo arquivo feito em vida sobre Tom, revela .

par=31820: Desconfiança -- Os arquivos divulgados na semana passada mostram que as acusações da freira foram recebidas com desconfiança pelos funcionários da embaixada na Cidade da Guatemala .

par=39746: Para eles se abrem os sésamos de nossos arquivos, em geral fechados para nós .

par=Esporte-94b-des-2: Kundts, do FBI, disse que o órgão é resposável pelo relacionamento com agências de segurança de outros países, troca informações com as polícias federais de todo o mundo e tem acesso a arquivos de imigração em várias localidades .

par=Especial-94a-nd-1: Pode discar para a Cut (Central Única dos Trabalhadores) ou simplesmente consultar os arquivos da SAE (Secretaria de Assuntos Estratégicos .

par=Especial-94b-nd-2: Quando o Ipes encerrou suas atividades, o senhor Clycon de Paiva, que o presidia na ocasião, deu instruções para que toda a documentação existente nos arquivos do Instituto fosse empacotada e entregue, sem expurgos, ao Arquivo Nacional .

par=19835: Na abertura dos arquivos do DOPS, Chico Buarque afirmou que, durante o regime militar, Paulo César teria dedado todo mundo .

>: Pesquisas nas áreas de historiografia, pensamento social, política, corporativismo e sindicalismo, educação, relações internacionais, metodologia de <STRONG< i>arquivos privados e legislação de arquivos

par=54380: Artigo 2 -- O Ombudsman parlamentar tem o direito de estar presente nas reuniões do governo, dos tribunais e órgãos adminsitrativos, tendo também acesso a arquivos do governo, dos departamentos governamentais, tribunais e de outras autoridades .

par=Brasil-94a-pol-1: Quando são conhecidos o número e a agência da conta, e este é o caso, seu levantamento não exige mais do que uma consulta ao arquivo de microfilmes do banco .

par=Brasil-94b-pol-3: A esses dirige-se o ponto do documento que pede o compromisso de abrir irrestritamente os arquivos da repressão política existentes sob sua jurisdição .

par=Dinheiro-94b-eco-1: Repousam nos arquivos do Itamaraty e do Ministério da Aeronáutica incidente diplomático grave, que poderia ter trazido sequelas ao relacionamento do Brasil com a Venezuela .

par=89322: III -- a autoridade, quando necessário, requisitará, para o exame, os documentos que existirem em arquivos ou estabelecimentos públicos, ou nestes realizará a diligência, se daí não puderem ser retirados ;

par=Esporte-94a-des-1: A Polícia Civil de Santo André apreendeu na noite de anteontem documentos, arquivos e objetos encontrados na sede da Mancha .

par=Brasil-94b-pol-1: O arquivo contra os adversários quercistas tem sido uma das poucas atividades dos escritórios de Quércia em São Paulo e Campinas (SP) .

par=Ilustrada-94b-nd-2: Para escolher as bases, recorreram ao arquivo doméstico .

par=Cotidiano-94a-soc-2: A quarta proposta, apresentada no Senado em 1989, diz que o pedido de informações pessoais constantes de arquivos públicos será deferido ou não no prazo de 48 horas .

par=Dinheiro-94b-eco-1: Mesmo quando um empresa é fechada, o DNRC mantém o nome da companhia e de seus sócios em seu arquivo .

par=Dinheiro-94a-eco-2: Na Receita Federal já devem estar no arquivo morto .

par=Esporte-94b-des-1: Os auditores fiscais estão recorrendo aos arquivos de cheques emitidos pelas empresas do esquema PC para saber a origem dos recursos recebidos por Farah .

par=Empregos-94a-eco-2: Quando as pessoas ouvem que eu sou uma ` fund raiser ' se surpreendem, diz Célia, que organizou um arquivo de 80 empresas com as quais mantém correspondência para a obtenção de fundos .

par=18300: O novo canal pretende utilizar todo o material de arquivo da Rede Globo, o que inclui as centenas de clipes já produzidos para o Fantástico e trechos de alguns musicais históricos, como os que uniram Elis Regina e Gal Costa ou João Gilberto e Rita Lee .

par=10247: Vamos usar cenas de arquivo, mostrar a Segunda Guerra, o campo de Auschwitz .

par=9232: Essas pessoas vão receber, em casa, uma carteirinha e, semestralmente, uma ordem de cobrança no valor de, no mínimo, R$ 30 ", explicou Tânia Leite, responsável pelo arquivo do cineasta David Neves, guardado no Tempo .

par=61642: Sob a coordenação de Telê Ancona Lopez, o projeto Archives da Unesco programou a edição crítica de doze títulos de autores brasileiros (15 b) ; Roberto de Oliveira Brandão e Dilea Zanotto Manfio já listaram mais de mil manuscritos literários existentes em bibliotecas, arquivos e coleções particulares; Enio Fonda dirige o Archivum Generale Poetarum Latinorum Brasiliensium que visa publicar e recolher as poesias latinas dos Escolásticos da Companhia de Jesus escritas desde 1896; um grupo de trabalho (GT) tendo como objetivo o estudo do manuscrito e a problemática dos arquivos foi formado na reunião da Associação Nacional dos pós-graduandos em letras e linguística (ANPOLL) em julho de 1990 em Recife .

par=37267: Unindo as diferentes partes deste texto, há uma linha tênue estendida na longa duração da história: a profunda semelhança entre as gravuras dos Desastres e quase todas as fotos da Guerra Civil que levantamos em arquivos e bibliotecas espanholas .

par=52912: Nos Arquivos Nietzsche, dirigido por ela, todo mundo falava de Mussolini, e foi com delírio que os Arquivos receberam a visista do embaixador da Itália, vido especialmente a Weimar para transmitir pessoalmente, à Elisabeth, os votos do Duce .

par=Cotidiano-94a-soc-2: Ontem, o supervisor Rodrigues foi à Delpom e ao Deic (Departamento Estadual de Investigações Criminais) tentar reconhecer os ladrões nos arquivos da polícia .

par=Ilustrada-94a-nd-2: Que não é de quadros e sim de clics feitos pelo próprio ao longo de sua vida ou colhidas em seu arquivo pessoal .

par=40756: O segundo conjunto de estruturas de dados consiste nos arquivos da memória de longo prazo, que representam a informação usada na geração de imagens .

par=Ilustrada-94a-nd-1: A coleção reúne cerca de 150 entrevistas com atores, atrizes e comediantes que estão divididas em 30 volumes ilustrados com várias fotos inéditas e de arquivo pessoal .

par=Esporte-94a-des-1: O suíço tem em seu arquivo mais de 200 mil slides de F-1, desenhou dezenas de cartazes de GPs e vendeu cerca de 300 quadros sobre o esporte .

par=Ilustrada-94b-nd-2: Ao pesquisar os arquivos de Joaquim Pedro, Monteiro descobriu que, além de O Imponderável Bento e Casa Grande e Senzala & Cia., o cineasta deixou quatro roteiros inéditos: uma adaptação de Buriti (de Guimarães Rosa) , outra de Grande Sertão: Veredas (sem Diadorim) misturada a um conto de Sagarana, Vida Mansa (baseado em uma idéia de Clarice Lispector) e Minas de Prata (inspirado em José de Alencar) .

par=Esporte-94b-des-2: E quem gosta de futebol vai guardar para o futuro, como um arquivo.

par=71908: 355: 2a. Lei 8.159, de 8.1.91 -- Dispõe sobre a política nacional de arquivos públicos e privados e dá outras providências (Lex 1991/12 e 106, ret .

par=Ilustrada-94b-nd-1: Uma única má notícia para quem gosta de televisão com elevadíssimo QI: não está disponível para distribuição por cabo o sinal da BBC World Service, uma espécie de CNN com sotaque britânico, com os arquivos de documentários da estatal inglesa mas sem a chatice dos talk shows da rede americana .

par=Ilustrada-94a-nd-1: O trabalho durou 15 meses e envolveu entrevistas com cem pessoas, pesquisa em jornais e revistas da época e no arquivo da cantora, doado ao Museu da Imagem e do Som do Rio .

par=Brasil-94b-pol-2: Ao estudar arquivos, ela descobriu que o pintor podia receber, em um só dia, até 20 kg de chumbo .

par=Esporte-94b-des-1: Além dos textos, este meu supermicro pode conversar diretamente com os arquivos da Fifa (que dão todas as informações estatísticas, oficiais e históricas sobre a Copa do Mundo) .

par=Especial-94a-nd-1: A única obrigação legal é manter em arquivo o nome dos contribuintes

par=Ilustrada-94a-nd-1: A coleção reúne cerca de 150 entrevistas com atores, atrizes e comediantes que estão divididas em 30 volumes ilustrados com várias fotos inéditas e de arquivo pessoal .

par=Esporte-94a-des-1: O suíço tem em seu arquivo mais de 200 mil slides de F-1, desenhou dezenas de cartazes de GPs e vendeu cerca de 300 quadros sobre o esporte .

Por cinco anos, os partidos e os candidatos deverão manter <STRONG< i>arquivos com as suas prestações de contas e a relação completa de todas as doações recebidas e a identificação dos doadores

par=36224: É a sua história contada através do arquivo pessoal de Mário de Andrade que acompanhou a sua trajetória com admiração singular .

par=Brasil-94b-pol-2: Nemércio Nogueira, assessor de imprensa de Quércia, nega a existência de arquivos contra os adversários quercistas .

par=Ilustrada-94a-nd-1: As fotos mais antigas e as reproduções de capas de jornal e de cartazes dos primeiros shows de Hebe são do arquivo pessoal da entrevistadora .

par=Especial-94a-nd-2: No governo Figueiredo foi para a PF e provou ser homem da mais absoluta confiança do regime militar: protegeu os arquivos do departamento e alguns policiais aparentemente ameaçados, na época, pela eleição do primeiro governador oposicionista de São Paulo .

par=52921: É verdade que Farias não pôde consultar o arquivo guardado pela família do filósofo -- por razões desconhelcidas e que parecem no mínimo suspeitas .

par=Cotidiano-94b-soc-1: A procuradoria confirmou que os bicheiros estão destruindo arquivos de contabilidade para evitar novas complicações com a polícia .

par=Cotidiano-94b-soc-2: Pode-se encontrar num mesmo arquivo, por exemplo, o caso da estudante A.P., 22, estuprada no início do ano, no campus da Universidade de Brasília (UnB) , por Ramon Oliveira da Silva, 18, e o de Francisca Neusa de Almeida Farias, 37, espancada e violentada pelo ex-marido, Glaucus Abi-Ackel, sobrinho do ex-ministro da Justiça, hoje deputado federal, Ibrahim Abi-Ackel (PPR-MG) .

par=Especial-94a-nd-2: A proposta na época era de se queimar e destruir os arquivos como forma de protesto contra a ditadura .

par=Cotidiano-94b-soc-1: Mas logo se viu que esse fenômeno era generalizado na região da Grande São Paulo. De acordo com as denúncias que estão nos arquivos das polícias Civil e Militar, os traficantes preferem usar meninas grávidas que, imaginam, seriam menos suspeitas .

par=Cotidiano-94b-soc-2: Além da falta de ares-condicionados, as obras estão amontoadas em estantes inadequadas para servir como arquivo .

par=114060: Não era uma casa pública, arquivo ou biblioteca, era um lugar onde, no que tocava a papéis e manuscritos, podia dar com alguma cousa privada e doméstica .

par=Fovest-94a-eco-2: Um historiador pode trabalhar em pesquisa acadêmica ou para particulares, em consultoria levantamento histórico em arquivos, organização de arquivos e ensino de primeiro, segundo e terceiro graus .

par=48320: A primeira vista, tudo parece indicar um desses trabalhos de preservação da cultura popular, encomendado por alguma instituição oficial de arquivo de imagens ou de elaboração da memória nacional .

par=Brasil-94b-pol-1: O computador pessoal do diretor do Banco Central, Gustavo Franco, era e continua sendo o arquivo secreto do plano .

par=Especial-94a-nd-2: Um projeto que beneficiaria os arquivos dos tribunais, abarrotados de papéis, 'tá parado no Congresso .

par=44374: Em primeiro lugar, eles ampliaram o arquivo de evidências empíricas que favorecem a existência e a flexibilidade da imagética mental .

par=Especial-94a-nd-2: A Resicop 'tá lançando uma linha de tubos-embalagens, destinada ao transporte e arquivo de mapas, desenhos, telas, posters e gravuras .

par=Cotidiano-94a-soc-2: As Forças Armadas possuem em seus arquivos dossiês sobre a situação do crime no Rio .

par=52890: Essa foi a primeira de uma série de providências que a levaram à propriedade e ao domínio exclusivo dos arquivos em que se encontravam os manuscritos e as cartas de Nietzsche .

par=Esporte-94b-des-2: Boa parte de todo o meu arquivo internacional, 70 anos de material, preenche as memórias de dois computadores .

par=Dinheiro-94b-eco-2: O arquivo do governo é feito a partir de informações colhidas em todos os cartórios do país .

par=Brasil-94b-pol-1: A julgar pelas notas lançadas nos arquivos da Escola Estadual Marcílio Dias, de Vicente de Carvalho, distrito do Guarujá (SP) , Lula foi um aluno muito bom na primeira série do primário .

par=40747: Em primeiro lugar, eles ampliaram o arquivo de evidências empíricas que favorecem a existência e a flexibilidade da imagética mental .

par=53704: Ele deverá ter livre acesso a todos os documentos e arquivos do Estado e, em caso de negligência ou desatendimento ao pedido do ombudsman por qualquer funcionário, o mesmo poderá sofrer uma multa de até mil coroas .

par=Esporte-94b-des-1: Mary Joe até pediu cópia de algumas reportagens para seu arquivo pessoal .

par=Esporte-94b-des-1: O consulado americano em São Paulo afirma que a autorização é apenas para que o FBI libere para o Comitê dados que eventualmente estiverem em seus arquivos .

par=Ilustrada-94b-nd-2: Um projeto que poderia atualmente se consagrar, graças aos recursos visuais existentes, como revolução jornalística numa emissora educativa, adormece nos arquivos da própria Cultura .

par=29599: Era um filme pacifista que retratava um brasileiro que lutou no Vietnã a partir de entrevistas e material de arquivo .

par=30875: Lisboa -- Proibidos durante 25 anos de vasculhar arquivos de quase meio século de ditadura, os portugueses discutem hoje no Parlamento o direito de acesso aos documentos que lhes dizem respeito .

par=Ilustrada-94b-nd-2: Parte das imagens sairá do arquivo pessoal de Gerald por exemplo: trechos de Sturmspiel, ópera que montou na Alemanha em 91 .

par=Especial-94a-nd-2: Eu era diretor do Deops (Departamento Estadual de Ordem Política e Social) e não guardador de arquivos, e é sobre isso que se trata .

par=Cotidiano-94b-soc-2: Estes laboratórios prestam serviços de restauração, reprodução e ampliação para arquivos, museus, empresas e clientes particulares .

par=Cotidiano-94b-soc-1: As técnicas de abordagem e a disposição dos inquéritos nos arquivos da Delegacia da Mulher também aproximam personagens ligados ao poder de criminosos anônimos .

par=Dinheiro-94a-eco-2: Uma folha de cheque custa para as instituições financeiras, em média, US$ 0,60 para o seu processamento e arquivo, enquanto a mesma operação de débito pelo cartão magnético cai para US$ 0,20, diz Márcio Santos Souza, diretor da administradora de cartões de crédito Bradesco Visa .

par=Brasil-94b-pol-1: Com os novos computadores, o Banco de Dados da Folha terá um arquivo de matérias e fotos que poderá ser acessado on line através de terminais da Redação .

par=Dinheiro-94a-eco-2: Sempre que o usuário quiser recuperar o desenho, chama o arquivo texto e o programa se encarrega de executar todos os comandos ali escritos .

par=29783: Por exemplo, copiar o arquivo para um disquete, é simples .

par=30563: -- A invasão do site da Embrapa (Empresa Brasileira de Pesquisa Agropecuária) por um garoto quedeletou arquivos importantes, foi considerada invasão de privacidade e depredação do patrimônio público .

par=Esporte-94b-des-1: O programa me permite, daqui dos EUA, pegar os arquivos sobre futebol no Banco de Dados aí da Folha, além de uma tabela com simulações das chances de cada time durante a Copa .

par=36168: Peterson diz ter analisado tanto arquivos de computador quanto os registros de entrevistas originais com os casais divorciados, arquivados agora no Murray Research Center no Radcliffe College .

par=Dinheiro-94a-eco-2: Além disso, como arquivos de imagem ocupam muito espaço em disco, o programa desenvolveu uma técnica habilidosa de arquivar em arquivo texto apenas os comandos que foram definidos pelo usuário .

par=30047: Ou seja, não adianta pôr o micro do lado da impressora e mandar imprimir um arquivo pois ele só será impresso se a impressora possuir um receptor Ir .

par=30261: O achado dos novos scanners de mesa é o reconhecimento ótico de caracteres (OCR) , uma tecnologia cada vez mais eficiente que permite transformar a imagem de um documento num arquivo editável .

par=Especial-94a-nd-2: Certifique-se de que a placa do micro poderá aceitar futuramente a conexão de dispositivos como uma placa de fax / modem (para envio e recebimento de mensagens e arquivos) e placas de som para 'cutar música no computador, por exemplo .

par=Especial-94a-nd-2: Ao criar um arquivo com cadastro de clientes, é possível consultar os dados por qualquer variável que você definir: nome, idade, telefone ou endereço, por exemplo .

par=48749: Um terminal de videotexto, por exemplo, pode ser inteiramente controlado na base de dados: ali são registrados todos os acessos aos seus arquivos, computados os tempos de acesso, tabuladas as preferências de cada usuário e os seus períodos de lazer e de trabalho, confrontando os resultados com a faixa etária, a condição econômica e o grau de instrução do indivíduo em questão .

par=Brasil-94a-pol-1: Esse processo foi radicalizado este ano com a implantação do primeiro arquivo de fotos em computador da imprensa brasileira .

par=25444: Como o sistema da Receita é fechado, não poderiam ser hackers-a denominação para designar os piratas que invadem arquivos de computadores, usando o acesso permitido por redes de comunicação, como a Internet .

par=Brasil-94b-pol-1: O erro foi cometido no momento de puxar o arquivo no computador, afirmou Nardon .

par=Dinheiro-94a-eco-2: Enquanto hits, como o PKZip, permitem comprimir arquivos exclusivamente em ambientes DOS, o Densus opera não só em DOS, como também em qualquer outro tipo de sistema operacional, permitindo que um arquivo seja enviado condensado de um sistema Unix para um PC .

ANNEX 2 - Sample Code *Arquivo*

```
def extract_features(line):
    features = {}
    if "arquivo" in line:
        index = line.index("arquivo")
    elif "arquivos" in line:
        index = line.index("arquivos")
    else:
        return features
    features["label"] = line[0]

    for i in range(1,3):
        try:
            features["w(-%d)" % i] = line[index-i]
        except:
            features["w(-%d)" % i] = "<None>"
        try:
            features["w(+%d)" % i] = line[index+i]
        except:
            features["w(+%d)" % i] = "<None>"
    return features

    if "computador" in line:
        features["computador"] = True
    else:
        features["computador"] = False

    if "computadores" in line:
        features["computadores"] = True
```

```
    else:
        features["computadores"] = False

    if "queima" in line:
        features["queima"] = True
    else:
        features["queima"] = False

    if "internet" in line:
        features["internet"] = True
    else:
        features["internet"] = False

    if "consulta" in line:
        features["consulta"] = True
    else:
        features["consulta"] = False

    if "pessoal" in line:
        features["pessoal"] = True
    else:
        features["pessoal"] = False

    if "internet" in line:
        features["internet"] = True
    else:
        features["internet"] = False

    if "DOS" in line:
        features["DOS"] = True
```

```
    else:
        features["DOS"] = False

    if "internet" in line:
        features["Windows"] = True
    else:
        features["Windows"] = False

    return features



def get_examples():
    featureset = []
    for i in range(1,4):
        if i == 3:
            f = open("corpus/arquivoSX.txt")
        else:
            f = open("corpus/arquivoS" + str(i) + ".txt")
        lines = f.readlines()
        lines = [tokenize(line) for line in lines]
        label = "sense" + str(i)
        for line in lines:
            feature = extract_features(line)
            featureset.append((feature, label))
    random.shuffle(featureset)
    return featureset



def get_scores(classifier, test_set):
    true_positives = 0
```

```
    false_positives = 0

    false_negatives = 0

    gold_standard = 0

    for (features, label) in test_set:

            guess = classifier.classify(features)

            if label == 'Yes':

                gold_standard += 1

                if guess == 'Yes':

                    true_positives += 1

                else:

                    false_negatives += 1

            else:

                if guess == 'Yes':

                    false_positives += 1

    accuracy = nltk.classify.accuracy(classifier, test_set)

    precision = float(true_positives)/(true_positives +
false_positives)

    recall = float(true_positives)/gold_standard

    return accuracy,precision,recall,true_positives,false_positives,
false_negatives
```