INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Extraction of Biographical Information from Wikipedia Texts

Sérgio Filipe da Costa Dias Soares
(Licensed in Information Systems and Computer Engineering)

Dissertation for the achievement of the degree:

Master in Information Systems and Computer Engineering

Committee

Chairman:	Prof. Doutor Luís Rodrigues
Main supervisor:	Prof. Doutor Bruno Martins
Co supervisor:	Prof. Doutor Pavel Calado
Observers:	Prof. Doutora Luísa Coheur

October 2011

Abstract

Documents with biographical information are frequently found on the Web, containing interesting language patterns and information useful for many different applications. In this dissertation, we address the challenging task of automatically extracting meaningful biographical facts from textual documents published on the Web. We propose to segment documents into sequences of sentences, afterwards classifying each sentence as describing either a specific type of biographical fact, or some other case not related to biographical data. For classifying the sentences, we experimented with classification models based on the formalisms of Naive Bayes, Support Vector Machines, Conditional Random Fields and voting protocols, using various sets of features for describing the sentences.

Experimental results attest for the adequacy of the proposed approaches, showing an F_1 score of approximately 84% in the 2-class classification problem when using a Naive Bayes classifier with token surface, length, position and surrounding based features. The F_1 score for the 7-class classification problem was approximately 65% when using the Conditional Random Fields classifier with token surface, length, position, pattern and named entity features. Finally, the F_1 score for the 19-class classification problem was approximately 59% when using a classifier based on voting protocols with length, position pattern, named entity and surrounding features.

Keywords: Sentence classification , Biographical Information Extraction

Sumário

Documentos com informações biográficas são frequentemente encontrados na Web, contendo tanto padrões linguísticos interessantes, bem como informações úteis para diversas aplicações. Nesta dissertação, abordamos a difícil tarefa de extracção automática de factos biográficos a partir de documentos textuais publicados na web. Para tal, segmentamos os documentos em sequências de frases, que serão classificadas como pertencendo a um qualquer tipo específico de facto biográfico, ou caso contrário, não relacionadas com factos biográficos. Para classificar essas frases foram usados diferentes modelos de classificação tais como, Naive Bayes, Support Vector Machines, Conditional Random Fields e protocolos de votação, utilizando diferentes conjuntos de características que descrevessem as frases.

Resultados experimentais comprovam a adequação das abordagens propostas, obtendo um resultado F_1 de aproximadamente 84% no problema de classificação em duas classes, ao usar o classificador Naive Bayes com base nas características das palavras, comprimento, posição e vizinhança das frases. Para o problema de classificação em sete classes foi obtido um resultado F_1 de aproximadamente 65%, ao usar o classificador Conditional Random Fields com base nas características das palavras, comprimento, posição, existência de expressões conhecidas e de entidades mencionadas. Finalmente, para o problema de classificação em dezanove classes foi obtido um resultado F_1 de aproximadamente 59%, ao usar um classificador baseado em protocolos de votação com base nas características de comprimento, posição, existência de expressões conhecidas e de entidades mencionadas, bem como a vizinhança das frases.

Keywords: Classificação de frases , Extracção de Informação Biográfica

Acknowledgements

My first acknowledgement goes to my parents (Abel Soares and Patrocínia Soares) and to my brother (Pedro Soares) for everything they do for me and for making my life a lot easier, specially during the most complicated moments and to provide me the opportunity to focus on my work.

I also want to thank to my supervisors (Bruno Martins and Pavel Calado) for their availability, their precious advices and in addition for all the papers they sent to me.

My closest friends, Luís Santos, Pedro Cachaldora and João Lobato with whom I could discuss some ideas and get a lot of important suggestions.

Professors Luisa Coheur and Andreas Whichert for answering to my help requests in the most complicated moments.

Professor Paulo Carreira for the most inspirational moments of my life, and for make me believe that everything is possible.

I also want to thanks to all my working neighbourhoods (João Fernandes, Nuno Duarte, José Rodrigues, David Granchinho, Ricardo Candeias, João Vicente, . . .) who provide me an enjoyable great place to work.

All my teachers who granted me the required knowledge to complete this dissertation.

Finally, I would like to express my most affective thanks to my girlfriend, Ana Silva, for her exceptional support, patience and dedication through almost all my university life and specially during the elaboration of this dissertation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the Web technology continues to thrive, a large number of documents containing biographical information are continuously generated and published online. Online newspapers, for instance, publish articles describing important facts or events related to the life of well-known individuals. Such Web documents describing biographical information often contain both meaningful biographical facts, as well as additional contents, irrelevant to describing the person (e.g., detailed accounts of the person's actions). Although humans mostly manage to filter the desired information, manual inspection does not scale to very large document collections.

For many applications, it would be interesting to have an automated system capable of extracting the meaningful biographical information from large document collections, such as those provided by news agencies. Moreover, it is our belief that, if relevant biographical information in human generated texts can be extracted automatically, this information can be used in the production of structured biographical databases, capable of supporting many interesting studies in the Humanities and other related areas.

Information Extraction (IE) is nowadays a well-established research area, with many distinct approaches for solving different kinds of IE problems having been described in the related literature. However, extracting biographical information from textual documents still presents many challenges to the current state-of-the-art.

Nevertheless, we have that several approaches for biographical information extraction were proposed in the field of IE. Some of those approaches leveraged on document retrieval (e.g., from the web and/or other repositories), considered documents of different kinds (e.g., structured, unstructured and semi-structured), and had different extraction objectives (e.g., personal information, related events, temporal information, etc.). Different systems also considered different

input cardinalities (e.g., some previous works focus on summarizing a single biographical document, while others summarize multiple documents into a single one), and produced different outputs (e.g., some systems produce a *templatized* output where the extraction program tries to fill a fixed set of template fields based on the input information, while other systems try to generate a plain text summary of the received documents).

In the context of this Msc thesis, we addressed the IE problem of automatically extracting meaningful biographical facts, specifically immutable (e.g., place of birth) and mutable (e.g., occupation) personal characteristics, relations towards other persons and personal events, from textual documents, through an approach based on sentence classification.

We propose to first segment documents into sentences, afterwards classifying each sentence as describing either a specific type of biographical fact, or some other case not related to biographical information. For classifying the sentences, we experimented with models based on the formalisms of Naive Bayes, Support Vector Machines, Conditional Random Fields and voting protocols, using various sets of features for describing the sentences.

Experimental results attest for the adequacy of the proposed approaches, showing an F_1 score of approximately 84% in the 2-class classification problem when using a Naive Bayes classifier with token surface, length, position and surrounding based features. The F_1 score for the 7-class classification problem was approximately 65% when using the Conditional Random Fields classifier with token surface, length, position, pattern and named entity features. Finally, the F_1 score for the 19-class classification problem was approximately 59% when using a classifier based on voting protocols with length, position pattern, named entity and surrounding features.

1.1 Hypothesis and Methodology

The objective of this work is to evaluate different approaches for automatically extracting biographical information from Portuguese Wikipedia's pages, focusing on football-related personalities. In order to perform the experiments, a corpus based on Portuguese Wikipedia documents was produced, based also on a proposed hierarchical taxonomy composed by 19 categories. Afterwards, a Java application was created, which relied on two different classification packages, namely Weka[1] and Minorthird[2]. This application allowed the realization of experiments using different feature groups and models to classify the given document's sentences, based on a set of training documents, with the objective of perceive how different feature groups and classifiers influence the classification results.

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://sourceforge.net/apps/trac/minorthird/wiki

1.2 Contributions

The main contributions of this thesis are as follows:

- The production of a hierarchical biographical taxonomy that covers the most common types of biographical facts.
- The creation of a Portuguese corpus based on Wikipedia documents referring to football celebrities, annotated with the referred taxonomy.
- A careful comparison of different classification algorithms and feature types applied to the classification of Portuguese sentences according to biographical categories.
- Development of a prototype system implementing the proposed methods, which was made available online as an open-source package on Google Code.
- The results reported on this dissertation were also partially published as article in the proceedings of the 15th Portuguese conference on Artificial Intelligence known as EPIA 2011.

1.3 Document Outline

The remainder of this report is organized in the following manner. First, Chapter 2 explains the fundamental concepts and surveys previous works in the area. Next, Chapter 3 presents my thesis proposal, detailing the considered features and classification algorithms. Chapter 4 describes the performed experiences as well as the obtained results. Finally, Chapter 5 presents some conclusions and points directions for future work.

Chapter 2

Concepts and Related Work

This chapter presents the core concepts in the area of Information Extraction, useful to fully understand the techniques used to accomplish my Msc thesis objectives, and also fundamental to understand other related works.

Next, this chapter presents previous works in the fields of Information Retrieval and Information Extraction, which attempted to address the problem of automatically extracting biographical information from textual documents.

2.1 Fundamental Concepts

First, it is important to realize that Information Extraction and Information Retrieval are distinct concepts often confused. In brief, while Information Retrieval concerns with identifying relevant documents from a collection, Information Extraction deals with the transformation of the contents from some input documents into structured data (Chang *et al.*, 2003).

Several authors have tried to define Information Extraction in the past (Cowie & Lehnert, 1996; Cunningham, 2005; McCallum, 2005). Based on those definitions, I hereby define Information Extraction as a set of techniques for deriving data from one or more natural language texts, which can be structured, unstructured or semi-structured, in order to extract snippets of information that can be redundant, ambiguous and erroneous, but hopefully are easier to analyze than the original documents.

Several authors have also tried to define Information Retrieval (Cowie & Lehnert, 1996; Cunningham, 2005). Based on those definitions, I hereby define Information Retrieval as the process of finding and presenting relevant documents to the user, from a large set of documents.

Both Information Extraction and Information Retrieval systems are typically developed and tested using a corpus, consisting on a large set of documents. In order to increase the usefulness of the corpus, the same is generally annotated in a process known as tagging, which consists in assigning appropriate labels to particular segments of the document's contents. This process can include, for example, the identification of dates, locations, person names, etc.

The process of tagging words in a document according to their morphological categories (i.e., verb, adjective, noun, etc.) is commonly referred to as part-of-speech tagging (Abney, 1996a). The part-of-speech tagger receives a group of words and a group of possible tags and assigns each word the most probable tag.

The most common types of structures extractable from unstructured texts written in natural language are entities, relationships and attributes. Entities generally consist of named entities, like people names, locations or temporal expressions, and can take the form of one or more word tokens. Relationships are associations over two or more entities. To distinguish between entity and relationship extraction tasks, Sunita Sarawagi argued that, whereas entities refer to a sequence of words in the source, relationships express the associations between two separate text snippets representing the entities (Sarawagi, 2008). The problem of relationship extraction can be generalized to N entities, and it is used, for instance, in the task of event extraction (Grishman & Sundheim, 1996). Finally, the objective of attribute extraction is to identify attributes associated with the entities, providing more information about them. This task is very popular in opinion extraction tasks, whose general objective is to analyze reviews to find out if a given entity has a positive or negative opinion associated with it (Pang & Lee, 2008). Note that there are many more types of extractable structures beyond the ones presented, including tables (Hurst, 2000; Liu *et al.*, 2007; Pinto *et al.*, 2003) or lists (Cohen *et al.*, 2002), but they will not be detailed nor used on this work.

The extraction of the above structures can be done at the level of sentences, typically for entity and attribute extraction, or using a broader scope like paragraphs or entire documents, generally used for relationship extraction. Moreover, one should consider that different types of document sources exist. In one hand, there are structured sources, like the computer generated pages supported by a database. The extractors for this type of sources are known as wrappers (Arasu & Garcia-Molina, 2003; Barish *et al.*, 2000; Baumgartner *et al.*, 2001), and the difficulty here is to automatically discover the template using the page structure.

On the other hand, there are unstructured sources, which are characterized by the lack of consistence and homogeneity. In this kind of source, the extraction is harder, but one can still exploit the scope of the documents to help the extraction task (e.g., news articles (Grishman & Sundheim, 1996), classified ads (Soderland, 1999), etc.). Furthermore, if the sources do not have an

associated scope, one can still exploit the redundancy of the extracted information across many different sources (Sarawagi, 2008).

Two different approaches are generally considered in information extraction tasks, namely rule-based approaches and machine learning approaches. In rule-based approaches (also known as hand-coded approaches), the developer defines rules, like regular expressions, that capture the desired patterns. This knowledge engineering approach requires for the developer to be very well informed about the content and format of the input documents, in order to create effective extraction rules that generate the desired result. Although this is the simplest approach, it does require a fairly arduous test-and-debug cycle in order to capture the desired patterns, and it is dependent on having linguistic resources at hand, such as appropriate lexicons, as well as someone with the time, inclination, and ability to write rules (Appelt & Israel, 1999).

Alternatively, it is possible to use machine learning techniques to automatically create the rules, based on a training corpus. Learning approaches use machine learning algorithms and an annotated training corpus, to train the system, avoiding the difficulties of writing extraction rules. Both, Appelt & Israel (1999) and Soderland (1999) argued that machine learning approaches allow an easier adaptation of existing Information Extraction systems to new domains, when compared with rule-based approaches. Note that, even when using this approach, it is essential to have domain expertise to label the training examples, and to define features that will be robust on unseen data (Sarawagi, 2008). In summary, rule-based approaches are easier to interpret and develop, being adequate in domains where human experts are available. On the other hand, machine learning approaches are more robust to noise in the unstructured data, being especially useful in open-ended domains (Sarawagi, 2008).

In general, different extraction systems use a similar information processing pipeline, in order to extract information from natural language texts. This pipeline represents a decomposition of the Information Extraction process in several tasks, giving more flexibility in the choice of techniques that better fit the objective of a particular application. The independence of each module also allows an easier debugging, and a customized extraction activity through reordering, selection and composition of techniques for the different subtasks. The referred pipeline is typically composed by the following steps: Segmentation, Classification, Association, Normalization and Correference Resolution (See Figure 2.1). The segmentation task divides the input text in segments or tokens. This task is relatively simple for the case of western languages due to the fact that, typically, the delimitators are a space or a punctuation character. However, in oriental languages, the task is harder and may require the use of external sources. The classification task determines the class of each segment. This classification can be made through rule-based or machine learning approaches. Machine learning is the most popular approach and generally uses supervised

Figure 2.1: Typical Extraction Pipeline (Gonçalo Simões, 2009)

techniques requiring an annotated corpus. The association task tries to discover relationships between the discovered entities (i.e., segments of text classified as belonging to a given entity type). The simplest approaches use rules to capture a limited set of relationships, while other approaches use syntactic analysis to exploit grammatical relationships (Grishman, 1997). Alternatively, machine learning approaches are also possible. For instance, probabilistic context-free grammars (Miller *et al.*, 1998), which have a probability value associated to each rule, can generate different syntactic trees, which have an associated probability. The most probable tree is then chosen. The normalization task is expected to standardize the information. This task usually is accomplished through the use of rules that convert the information into a specific format (e.g., convert temporal expressions to calendar dates). Finally, the correference resolution task tries to discover when one entity is referenced differently along the text. Both rule and machine learning approaches are commonly used to accomplish this task (Gonçalo Simões, 2009).

Although Information Extraction has more than two decades of research, it still presents many challenges. The main difficulties are related with the inability to develop a system fully capable of understanding the content of a given text, which leads to the usage of generic probabilistic approaches to deal with existing natural language processing problems. In brief, the performance of those systems is strongly dependent on statistics collected from training data, and consequently, the quality of the results is still far from the desired.

2.2 Related Work

Different approaches for biographical information extraction have been described in the past. Those approaches can be categorized into three kinds of systems that are relevant to this work, namely question answering systems, summarization systems and extraction systems.

2.2.1 Question Answering Systems

Typically, the more information we have, the longer it takes to find the wanted information. For instance, answering to the question *Where did Fernando Pessoa die?* can become a highly time-consuming task. The approach taken to answer those questions generally consists of using a web search engine, thus requiring the choice of the indicative keywords in order to retrieve a vast set of documents considered relevant by the search engine. Silva et al. argued that question answering (QA) systems deal with these problems by providing natural language interfaces, where the users can express their information needs, and retrieve the exact answer to the posed questions, instead of a set of documents (Silva, 2009).

Ideally, a QA system should be capable of answering any given natural language question based on a given source of knowledge. The first approaches to building QA systems consisted on writing some basic heuristics and rules (typically exploiting words like, who, when, where, etc.) for question classification and, consequently, required too much time and effort and resulted in systems that were costly and also not adaptable to different domains or languages (Kwok *et al.*, 2001). In order to solve the referred problems, the current trend in QA is the usage of machine learning techniques, allowing the system to automatically learn rules and avoid the need of having a human expert handcrafting them (Li & Roth, 2002). Moreover, the development effort decreases and different domains can then be covered.

The general architecture of a QA system involves three distinct components, namely question

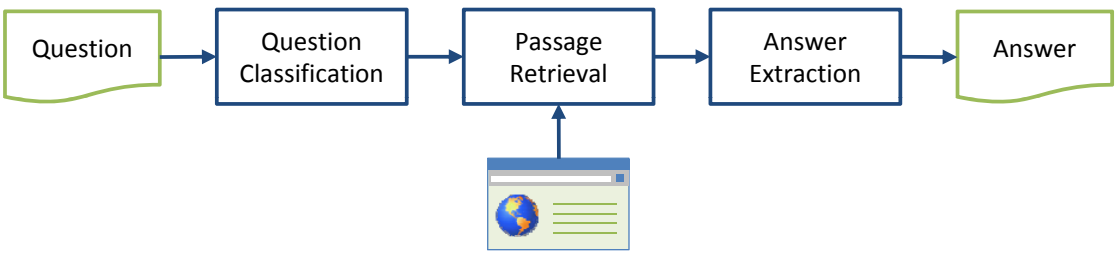

Figure 2.2: General Architecture of a QA System (Adapted from (Silva, 2009))

classification, passage retrieval, and answer extraction (See Figure 2.2).

This division allows an easier comparison between different solutions and allows the improvement of any component without affecting the others. The question classification component should determine the question's category, since the question and answer categories are strongly related. The passage retrieval component finds relevant information (e.g., candidate sentences) from a pre-defined knowledge source. Finally, once the question category is known, and once relevant information is possessed, the objective of the extraction component consists of selecting the final answer based on the existing candidate answers.

2.2.1.1 Question Classification Module

Moldovan et al. argued that about 36.4% of the errors in a QA system are caused by this module (Moldovan *et al.*, 2003). The objective of this module is to determine some of the constraints imposed by the question on the possible answer and discover the type of answer expected, through the discovered semantic category of the question.

Li and Roth defined a taxonomy of 6 coarse and 50 fine grained classes (Li & Roth, 2002), which are widely used in the question classification task, although several other question type taxonomies have been proposed in the literature (Hermjakob *et al.*, 2002). Silva et al. claimed that depending on the question category, different processing strategies can be chosen to find an answer. For instance, Wikipedia can be used for questions classified as Description:Definition. Furthermore, knowing the question's category can restrict the possible answers (Silva, 2009).

Several authors also tried different alternatives, achieving new state-of-art results (Blunsom *et al.*, 2006; Li & Roth, 2002; Pan *et al.*, 2008). However, the actual state-of-art accuracy result in question classification was achieved by Silva et al. whose work is described below.

Silva et al. addressed the task of question classification (QC) as a supervised learning problem, with the objective of predicting the category of unseen questions. In order to accomplish the described task, he tested a rich set of features that are predictive of question categories, in order to discover the subset which yields the most accurate results. Those features are word level n-grams, question headword, part-of-speech tags, named entities and semantic headwords. Silva et al. tested the above features with three different classification algorithms, namely Naive Bayes, the k-nearest neighbors algorithm (k-NN) and Support Vector Machines (SVMs), which yielded the best results. The best accuracy in his work was 95.4% for coarse-classification and 90.6% for fine-grained classification through the use of the question headword, semantic headword and unigrams (n-grams in which $N = 1$). This approach represents the current state-of-art. Table 2.1 shows a comparison of previous works on question classification, which used the same taxonomy

Author	Year	Coarse Granularity	Fine Granularity
Li & Roth	2002	91.0%	84.2%
Zhang et al.	2003	90.0%	80.2%
Hacioglu et al.	2003	—	80.2 - 82%
Krishnan et al.	2005	93.4%	86.2%
Blunsom et al.	2006	91.8%	886.6%
Pan et al.	2008	94.0%	—
Hung et al.	2008	93.6%	89.2%
Fangtao et al.	2008	—	85.6%
Silva et al.	2009	95.2%	90.6%

Table 2.1: Comparative of Question Classification Systems

and the same training and test sets.

2.2.1.2 Passage Retrieval

Several approaches can be used in passage retrieval, and also a different approach can be used for each different question category. For instance, in one hand, Google's search engine could be used to extract snippets that contain the answer for a factoid-type question. On the other hand, encyclopedic knowledge sources such as Wikipedia or DBPedia could be used to answer non-factoid questions (e.g. definitions, etc.) that require longer answers. However, many approaches for optimal query creation for the web search engines were created, in the context of QA systems. For instance, Oren Tsur et al. handcrafted a set of features (such as "born", "graduated", "suffered", etc.) that could probably trigger biography-like snippets when combined with the target of the definition question, as a query to the web search engine (Tsur *et al.*, 2004). Other authors tried a more naive approach to query formulation, by sending the whole question to the IR system. However, this approach is not very effective, because IR systems are not capable of understand natural language questions, and also because IR systems ignore the stop words and often stem the query terms, consequently eliminating the user's intention. Since no perfect query format was discovered, many undesired documents are always returned. Thus, text classification techniques are required to classify the retrieved documents in order to filter the irrelevant ones. Some authors used probabilistic classifiers, and then use the documents classified with the desired class to extract snippets of text that may compose the answer. Oren Tsur et al. compared two text classifiers, Ripper and SVMs, for their QA system, demonstrating the benefits of integrating them to filter search engine results (Tsur *et al.*, 2004). Other authors used external sources of knowledge, such as Wordnet, to improve system's performance and coverage.

Another approach for query composition consists in rewriting the question. This approach exploits the fact that the Web has plenty of redundancy and, consequently, the answer to a question should exist on the web written differently. Some authors have developed algorithms which learn question rewrite rules. These algorithms receive a question-answer seed pairs to learn new question rewrite patterns, and validate each learned pattern with a different question-answer pairs in order to remove incorrect patterns. This technique allows the extraction of valuable candidate answers from the returned web search engine results.

2.2.1.3 Answer Extraction

The candidate answer extraction can leverage on the knowledge about the respective question classification. Thus, different strategies for answer selection can be used, based on the question's classification. For instance, in Numeric type questions, Silva et al. developed an extensive set of regular expressions to extract candidate answers (Silva, 2009). Furthermore, a gazetteer can be used in certain question categories, such as the Location:Country or Location:City categories, since they have a very limited set of possible answers.

After choosing the set of candidate answers, it is possible to filter some of then. Silva et al. implemented a filter which removes candidate answers, which are contained in the original question (Silva, 2009). Liang Zhou et al. proposed a filtering phase that deleted direct quotes and duplicate phrases (Zhou *et al.*, 2005).

At last, the final answer must be chosen. Several techniques exist to support this decision. Mendes et al. assumes that the correct answer is repeated on more than one text snippet, and thus the returned answer is the most frequent entity that matches the type of the question (Mendes *et al.*, 2008). Silva et al. grouped together similar candidate answers into a set of clusters. Next, he assigned a score for each cluster, which is simply the sum of the scores of all candidate answers within it. Finally, the longest answer within the cluster with the highest score is chosen as the final answer (Silva, 2009).

2.2.2 Summarization Systems

The objective of automatic summarization systems is to simulate the human production of resumes, although the state-of-art results are still far from accomplishing this. The resulting document should correspond to a small percentage of the original, and yet it should be just as informative (Zhou *et al.*, 2005). Kupiec et al. stated that document extracts of only 20% can

be as informative as the original one (Kupiec *et al.*, 1995). Rath et al. concluded that the optimal extract is far from being unique, and also that little agreement exists between summaries produced by persons and machine methods (based on high-frequency words) in the selection of representative sentences (Rath *et al.*, 1961). Kupiec et al. argued that summaries can be used as full document surrogates or even to provide an easily digested intermediate point between a document's title and its content, which is useful for rapid relevance assessment (Kupiec *et al.*, 1995). Luhn argued that the preparation of a summary requires not only a general familiarity with the subject, but also skill and experience to bring out the salient points of an author's argument (Luhn, 1958). Brandow et al. argued that to achieve human-like summaries, a system must understand the content of a text, correctly guess the relative importance of the material and generate coherent output (Brandow *et al.*, 1995). Unfortunately, all of those requirements are currently beyond the state of the art for anything more than demonstration systems or systems that are highly constrained in the domain.

There are two types of summarization systems, namely (i) single-document summarization (SDS) systems, which summarize only one document at each time, and (ii) multi-document summarization (MDS) systems, which receive two or more documents and summarize them into just one. MDS systems are more complex than SDS systems because the techniques of extract-and-concatenate used on SDS systems do not respond to the problems of coherence, redundancy, co-reference, etc.. In addition, while the sentence ordering for SDS can be the same as that of the original document, sentences extracted by a MDS system need a strategy on ordering to produce a fluent summary. Besides that, the input documents can be written by different people with distinctive writing styles, resulting in an additional problem. Mani argued that biographical MDS represents a substantial increase in system complexity and is somewhat beyond the capabilities of present day MDS systems (Mani, 2001). His discussion was based, in part, on the only known MDS biography system at that time (Schiffman *et al.*, 2001), which used corpus statistics along with linguistic knowledge to select and merge data about persons.

Furthermore, both SDS and MDS systems can be classified by their summary types, namely, (i) generic summaries, when they try to resume any type of given document, or (ii) special-interest summaries, which consist of document summaries based on a predefined topic.

Beyond that, summaries can be classified as informative, indicative or critic, in relation to the function they perform. The informative summaries can dispense the reading of the source-document. Contrarily, the indicative summaries only give an idea about the original's document content. Finally, the critic summaries present opinions about the expected content (e.g. book reviews).

Furthermore, (Mani, 2001) stated that summaries can take the form of an extract or an abstract. The extract form consists of extracting a subset of the original document data that is indicative

Categories	Classification
Number of Documents	Single-Document Summarization (SDS) Multi-Document Summarization (MDS)
Audience	Generic Summaries Special Interest Summaries
Summary Classification	Informatice Indicative Critic
Summary Form	Extract Abstract

Table 2.2: Different types of summarization systems

of its contents, through the use of linguistic analysis and statistics in order to keep its meaning. The abstract form consists of condensing information through knowledge-based methods, originating document contains materials not present in the input documents. This distinction arises due to the difficulties in the production of a coherent narrative summarization because it involves discourse understanding, language generation, and other complex natural language problems (Kupiec *et al.*, 1995). Nonetheless, some abstract-like summarization systems have had some success in restricted domains like highly structured technical papers (Paice & Jones, 1993), financial news (Jacobs & Rau, 1990) and others (Reimer & Hahn, 1988; Tait, 1985). However, the simpler and more general summarization approach is the one that consists only on the extraction task (Luhn, 1958), which avoids the referred natural language processing problems and focuses on making the best selection of representative document extracts. In brief, Table 2.2 resumes the different types of summarization systems presented above.

A MDS system starts with some documents and divides them into segments, typically sentences, which are then assigned to a predefined class by a classifier. For biography summarization systems, the classes should correspond to the most frequent attributes shared among biographies. For instance, (Zhou *et al.*, 2005) identified the following common biographic attributes: bio (birth and death), fame factor, personality, personal, social, education, nationality, scandal, and work.

Zhou et al. designed two classification tasks, one considers 10-classes and the other considers 2-classes. In 10-class, each input segment is assigned to the class with the same name of their most predominant class annotation. If the segment has no annotation it is assigned to a class called “Others”. In 2-class, each input segment with an annotation is assigned to a “bio” class and all the other to an “Others” class. Thus, two different classification granularities were compared.

After defining the candidate set of classes, the next step is training a classifier capable of assigning each segment, of any input document, to a referred class. Zhou et al. used a corpus of 130

biography documents, in which 100 documents were used to train their classification module and 30 documents were used to test it.

For their classification module, they experimented with three machine learning methods for classifying the sentences, namely, Naive Bayes, Support Vector Machines and C4.5 Decision Threes. The resulting F_1 score was of 82.42% with Naive Bayes, 75.76% with Decision Threes and finally, 74.47% with SVMs, on the 2-class classification task.

After the previous step, each segment belongs to a class assigned by the classifier. In other words, we have several segments for each important class of information that should be referred on any biography document. Zhou et al. argued that the summarization process can be guided using checklists, especially in the field of biography generation because a complete biography should have at least a fixed set of aspects of a person's life (Zhou *et al.*, 2005). Thus, fulfilling the biography checklist can be seen as a classification problem, in which one or more segments of each class should be present in the resulting document.

Alternatively, other authors (Schiffman *et al.*, 2001) tried a different approach for developing a summarization system. Instead of using a classifier to label each segment with a predefined class, they used the Alembic part-of-speech tagger (Aberdeen *et al.*, 1995) as a sentence tokenizer, the NameTag named entity tagger (Krupka, 1995) and the CASS parser (Abney, 1996b), obtaining a set of tagged sentences, which should be analyzed by a finite state machine in order to identify pre- and post- modifying apposite phrases, since appositive phrases and relative clauses typically contain important biographical information. Next, they used a cross-document co-reference program from the Automatic Content Extraction (ACE) research program, which compares names across documents based on similarity of a window of words surrounding each name, and which consider different ways of abbreviating a person's name (Mani & MacMillan, 1996). Thus, the set of descriptions found for each person in the collection are grouped together.

Regardless the chosen approach, in both cases, the result consists of a set of raw sentences, which need to be filtered before the choice of which sentences should appear on the final document. (Zhou *et al.*, 2005) filtered sentences by deleting direct quotes, dialogues, and segments with less than 5 words. He also merged sentences classified as biographical with the name-filtered sentences and finally deleted the duplicates. (Schiffman *et al.*, 2001) deleted duplicate appositive phrases as well as phrases whose headword does not appear to refer to a person. WordNet was used to check if the head word referred to a person.

In addition of the cleaning phase, Schiffman et al. used a merging approach also based on WordNet. If the descriptions have the same head stem, or both heads have a common parent below Person in WordNet or, even if one head subsumes the other under Person in WordNet,

then the descriptions are merged, being chosen as merged head the more general frequent head. Furthermore, when a merge occurs, the most frequent modifying phrase that appears in the corpus with the selected head is used. At last, if a person has more than one description with distinct modifiers, they are conjoined together, so that, *Wisconsin lawmaker* and *Wisconsin democrat* yields *Wisconsin lawmaker and democrat*. Finally, as a consequence of this cleaning and merging steps, a document with a set of biographical sentences is obtained. Thus, the remaining processing will be executed over this document, and a subset of it will constitute the document presented as the result. Typically, the next phase consists on ranking the remaining sentences in order to choose the best subset. Many authors ignore the previous steps, except the segmentation step, and continue directly to the ranking step, described below.

As already described, the remaining sentences should be ranked in order to make a wise choice of which sentences should remain in the final document. As a result, a document is obtained containing all accepted sentences ordered by an order parameter (Mani, 2001). Several authors (Zhou *et al.*, 2005) used the inverse-term-frequency (ITF) as a metric to estimate the information value. Thus, words with low frequency have a high value and other with high frequency have low value, allowing the identification of passages that are unusual in texts about the person.

Finally, top scoring sentences would be analyzed to remove redundancy. For instance, (Zhou *et al.*, 2005) modified the algorithm originally proposed by (Marcu, 1999) so that an extract can be automatically generated by starting with the sentences classified with the highest score by the ITF method and systematically removing a sentence at a time as long as a stable semantic similarity with the original text was maintained. This redundancy elimination phase was repeated until the desired summary length is achieved. (Brandow *et al.*, 1995) claims that the issues in increasing the relevant content of the produced summary are mainly in sentence selection heuristics. He also argued that more sophisticated techniques for discovering the important words (morphology, name recognition, etc.) would not contribute significantly to improve the choice of sentences.

A different approach was attempted by (Brandow *et al.*, 1995), which used some predefined key phrases in order to discover important extracts to include in the summary (e.g. *the objective of this study is*, *the findings from our research show*, etc.). Unfortunately, they realize that those phrases are strongly source document-dependent and cannot be generalized across the entire range of documents to summarize. However, this technique can be useful in the biography domain due to the existence of typical structure of biographic sentences (e.g. *... was born in...*, *...has lived in...*).

By their side, Edmundson's extraction system assigned numerical weight like ITF method, but based on other machine-recognizable characteristics or clues (Edmundson, 1969). Thus, the resulting score of each sentence was given by the sum of the four basic methods, namely cue, key, title and location (See Table 2.3).

		Structural Sources of Clues	
		Body of Documments	Skeleton of Documents
Linguistic Sources of Clues	General Characteristics of Corpus	Cue Method	Location Method
	Specifics Characteristics of Documents	Key Method	Title Method

Table 2.3: Four Basic Methods of Edmundson's Research (Adapted from (Edmundson, 1969))

The cue method assumes that the relevance of a sentence is affected by pragmatic words such as "significant", "impossible", etc. Through the use of a given cue dictionary with appropriate words and the respective cue weight, the sentence weight is the sum of the cue weights of its constituent words. The keyword method follows the same principle of the one proposed by Luhn for creating automatic extracts, which assumes that high frequency content words are relevant (Luhn, 1958). Thus, the weight of a sentence is the sum of key weight of its constituent words. The title method has in account the characteristics of the document's skeleton, for instance, title and headings. This method assumes that the document's author made a wise choice of document's title so it can reflect the subject matter of the document. Furthermore, this method assumes that the author, when partitioning the document into sections, also summarized those sections by choosing a descriptive heading. Thus, all meaningful words on the title receive a positive weight and consequently, the weight of each document sentence is the sum of the title weights of its constituent words. Finally, the location method gives special attention to document's format and headings.

Edmundson explained that this method is based on the hypothesis that sentences occurring under certain headings should be relevant, and also that relevant sentences tend to occur very early or very late in a document and its paragraphs. Thus, the method uses a prestored heading dictionary of selected words in the corpus that appear in the headings of documents. Next, the position method assigns positive weights provided by the heading dictionary and also positive weights to sentences based on their position in the text. At last, the weight for each sentence is the sum of heading weight and its ordinal weight. In brief, the relative weights of the four basic methods can be parameterized by the following function:

$$a_1 C + a_2 K + A_3 T + A_4 L$$

In the function, a_1, a_2, a_3 and a_4 are positive integers representing the weights of Cue, Key, Title and Location, respectively.

Later tests concluded that the cue-title-location method had the best score, while the key method

Feature	Individual Sents Correct	Cumulative Sents Correct
Paragraph	163 (33%)	163 (33%)
Fixed Phases	145 (29%)	109 (42%)
Leght Cut-off	121 (24%)	217 (44%)
Thematic	101 (20%)	209 (42%)
Uppercase Word	100 (20%)	211 (42%)

Table 2.4: Comparative of Kupiec's et al. Features Performance (Adapted from (Kupiec *et al.*, 1995))

in isolation had the worst score. (Edmundson, 1969) concluded that although keywords are important for indexing, they may not be useful for extraction. Furthermore, he does not consider keywords important for an extraction system because avoiding frequency counting over all the document results in a system simplification and shorter running time. In his work's conclusion, Edmundson alerted that future automatic abstracting methods must take into account syntactic and semantic characteristics instead relying simply upon gross statistical evidence.

A similar approach was undertaken by Kupiec et al., which used a simple Bayesian classifier to assign a score for each sentence in order to select the best sentences for inclusion in the generated summary (Kupiec *et al.*, 1995). The referred classifier used five features, namely sentence length cut-off, fixed-phrase, paragraph, thematic word and uppercase word feature. The sentence length cut-off feature assumes that short sentences are not important, and consequently, given a threshold (in his work, the threshold was set to 5) the feature is true only for sentences longer than the threshold, and false otherwise. The fixed-phrase feature assumes that the existence of some well-known sequences of words or being right under any title or heading containing some keywords are indicative of an important sentence. The paragraph feature assumes that the first and last paragraphs are generally important. Thus the paragraph's sentences are distinguished through the location of the paragraph where they are. The thematic word feature consists of selecting the thematic words which are the most frequent content words. Next, each sentence is scored as a function of existing thematic words. The highest score sentences are scored as true and others as false. At last, the uppercase word feature assumes that very frequent names are important in the summary. Thus, the sentences containing any frequent uppercase word (which is not an abbreviated unit of measurement) are classified as one and the remaining as zero. Sentences in which such uppercase words appear first score two. The referred Bayesian classifier was trained with a corpus created by professional abstracters by reference to the original document. Thus, the classification function learns to estimate the probability of a given sentence being included in an extract summary. Consequently, new extract summaries can then be generated for unseen documents through sentence ranking according to their resulting probabilities. Table 2.4 shows a comparison of the features used by (Kupiec *et al.*, 1995).

In the table 2.4, the first column refers to the feature, and the second and third columns show the sentence-level performance for individual feature and how performance varies as features are successively combined together, in descending order of individual performance, respectively. The best performance is given by the combination of paragraph, fixed phrase and sentence-length features. Furthermore, a slight decline in performance is observable when the thematic and uppercase features are also used.

Several other authors, like (Brandow *et al.*, 1995), tried the same approach through the use of a different set of features, in order to choose the best possible sentences, for instance, the presence of signature words in the sentence (which identifies words in a document that are relatively unique to that document), its location in the document, the target length of the extract, the type of extracting to be generated, etc.. Furthermore, to enhance readability, Brandow et al. added to the summary single sentences which separate sequences of sentences containing signature words. Moreover, the first one or two sentences of a paragraph are also added to the summary when the second or third sentence of the paragraph contains signature words.

(Schiffman *et al.*, 2001) used an alternative method for choosing the most valuable sentences. They used statistical methods in order to find out promiscuous verbs (weakly associated with many different subjects), and to discover which verbs are strongly associated with particular roles. For instance, the role police is typically associated with the verbs confiscate, shoot, arrest, etc.. Consequently, sentences where the main verb is a promiscuous one should be penalized, and sentences which main verb is highly associated with the target's role should be chosen to stay in the final biography document in, detriment of the remaining ones.

In brief, many heuristics were proposed to guide the choice of text extracts (Edmundson, 1969; Luhn, 1958), such as the presence of high-frequency content words (keywords), pragmatic words (cue words), title and heading words, structural indicators (sentence location), etc.. Sadly, there are no clear winners, although evidence suggests that the combination of some heuristics has the best performance (Edmundson, 1969). Thus, many possible summaries can exist and the decision of which is the best is still a non trivial task. Beyond that, Brandow et al. alerted that the creation on document resumes through the choice of highly informative sentences could seriously affect the readability, consequently reducing the summary utility.

2.2.3 Extraction Systems

Biographical extraction systems aim to classify sentences into categories related to biographical information. Biographical extraction systems can be integrated with other systems. For instance, they can be used to retrieve all the sentences classified as biographical, which could then be

used to answer biographical questions in QA systems or to generate a biography automatically in a summarization system. Thus, several extraction techniques were already addressed on Section 2.2.1 and 2.2.2. However, this section elaborate more on the extraction task, and covers different techniques not referred on the QA or summarization sections.

Conway approached the problem of biographical sentence classification as a genre classification task. Consequently, different approaches to genre classification were discussed in his work (i.e., Systemic Functional Linguistics and Multi-Dimensional Analysis) followed by a brief overview of stylistics (formal study of literary style), in which stylistic features revealed themselves as important for classifying biographical sentences (Conway, 2007). The concept of style can be defined as *the aesthetic "residue" when propositional content has been removed* and refers to the choice of expressions associated with a given genre or writer. His discussion focused on several problems such as what do people do with language, and how do they do it (systemic functional grammar), identifying features, which allow the differentiation of texts (Stylistic Analysis), authorship attribution (stylometrics), etc.. Moreover, he referred that a biography, as a genre, is the history of the lives of individuals, based on facts, and retaining a chronological sequence.

Additionaly, one should have in consideration that biographies present information using a journalistic style known as the "inverted triangle" (Pape & Featherstone, 2005). In the case of biographies, the inverted pyramid is typically composed by four parts. The first part, namely introduction paragraph, should contain the following essential attributes: birth and death dates, location of birth, profession, notable achievements, significant events, and optionally the marital status, and details of children and parents. The second part, namely expansion paragraph/s, should expand the initial facts using a narrative structure and usually chronologically. The third part, namely background paragraph/s, should provide relevant background information. In the last part, namely conclusion, it should be presented a graceful conclusion. However, there are exceptions such as in short biographies, which typically include the described first part, or literary biographical texts where the "inverted pyramid" model is likely not to be applied.

Furthermore, many resources are reproduced online, and numerous websites generate biographies of different lengths and domains. Possibly, the most common biographical subtype is the obituaries, which focus on the achievements, and typically do not refer the cause of the death (Ferguson, 2000).

Automatic text classification can be seen as the automatic assignment of texts to predefined categories. There are two types of categorization, namely endogenous, which is the main focus of today's research, being entirely based on the contents of the document itself, and exogenous, based on a document augmented by metadata. Furthermore, classification tasks can be distinguished with respect to overlapping categories, which allow the items to be classified in more

than one category, while non-overlapping categories restrict the assignment to only one category. For instance, binary classification constitutes a special case of non-overlapping categorization, where the number of categories is limited to two. Moreover, items can also be assigned to categories with a certain degree of probability.

In his work, Conway tested the possibility of reliable identification of biographical sentences. In order to do that, the corpus used was distinguished in two types: biographical corpora and multi-genre corpora, both written in english. The biographical corpus used was composed by texts from: The Oxford Dictionary of National Biography, Chambers Biographical Dictionary, Who's Who, The Dictionary of New Zealand National Biography, Wikipedia's biographies, a biographical corpus developed at the University of Southern California and TREC news text corpus. The multi-genre corpus used in the experiments was the BROWN corpus and the STOP corpus.

Then, an annotation scheme for classifying biographical sentences was created, and included six tags, namely, key (key information about a person's life course: date of death and birth, nationality, etc.), fame (what a person is famous for, both positively or negatively), character (attitudes, qualities, character traits and political or religious views), relationships (both with family and friend, etc.), education (attitudes, qualities, character traits and political or religious views) and work (positions, job titles, affiliations, etc.). Finally, a small corpus was created using the new annotation scheme, and included 80 documents from four different sources.

In order to perceive if people are able to reliably distinguish between isolated (that is, context-less) biographical and non-biographical sentences, a human study was performed. The study conducted showed that the participants had some confusion over the distinction between the core biographical (birth and death dates, education history, nationality, etc.) and extended biographical (not directly about that individual) classes. Consequently, a new study was conducted under the same conditions, except that the considered classification categories were only biographical (key, fame, character, relationships, education, work) and non biographical (unclassified). The results showed that good agreement can be obtained between multiple classifiers using the binary classification scheme developed to classify each sentence of the set as biographical or not. Moreover, the data gathered in the main study (500 sentences with a high agreement) was used subsequent machine learning experiments in order to assess the accuracy of automatic sentence classification. However, the referred data, known as "gold standard data" used is derived from the researcher's annotation efforts rather than those of participants involved in the study.

Thus, Conway concluded that people are able to reliably identify biographical sentences, and decided to test if it can also be performed automatically. Consequently, he selected a set of learning algorithms to accomplish the task of automatic identification of biographical sentences, namely, the Zero Rule (as baseline), the One Rule, C4.5 Decision Trees, Ripper Rules, Naive

Algorithm	Mean Accuracy(%)	Standard Deviation
ZeroR	53,09	0,95
OneR	59,94	4,85
C4.5	75,87	5,85
Ripper	70,18	6,33
Naive Bayes	80,66	5,14
SVM	77,47	5,62

Table 2.5: Learning Algorithms compared by (Conway, 2007)

Bayes and Support Vector Machines (described in the original report by (Conway, 2007)).

The feature used was the most common 500 unigrams from the Dictionary of National Biography (DNB), because it contains a large number of function words as well as biographical indicative words ("died", "married", etc.). Conway explained that the features were not extracted from the gold standard data to avoid the possibility of artificially inflating classification accuracy. The main objective of this experiment was to allow the extraction of indicative results about the usefulness of different machine learning algorithms for the biographical sentence classification task. Consequently, no further feature selections were used because the objective of the experiment is to compare different learning algorithms using a constant feature set. The reason that motivated this study was that although several published work comparing feature sets for genre classification exists, little work as focused on the comparison of learning algorithms. For instance, (Zhou *et al.*, 2005) used SVM, C4.5, and Naive Bayes for biographical sentence classification, but the focus of his work was the identification of optimal features for biographical classification, rather than comparing different learning algorithms (work described on Section 2.2.2).

Thus, Conway's work differed from the work by (Zhou *et al.*, 2005) in that it explores the performance of six algorithms instead of only three, also using a different data set and used a feature set composed of the five hundred most frequent unigrams of the DNB. The results obtained for the 10 x 10 fold cross validation run for each algorithm (See Table 2.5) revealed that the Naive Bayes algorithm obtained the best accuracy (80.66%), followed by the SVM algorithm (77.47%) and then C4.5 (75.87%).

The Naive Bayes algorithm performed better than all the other algorithms (except SVM) at a statistically highly significant level, when subjected to the corrected re-sampled t-test. Moreover, Naive Bayes performed better than the SVM algorithm, although not at a statistically significant level, because it failed to meet the significance threshold $P = 0.1$. In order to test the reliability of the results, the experiment was repeated using 100 x 10 fold cross validation, resulting in a mean score difference for each algorithm of less than 0.5%. This result confirmed the report of

(Bouckaert & Frank, 2004), which argued that 10 x 10 fold cross validation, in conjunction with the corrected resample t-test, allows reliable inferences concerning classifier performance.

Although many differences exists between the Zhou et al. and Conway's work, both concluded that Naive Bayes obtained the best results which also was remarkably similar (82.42% and 80.66%), also confirming the results reported by other authors (Lewis, 1992). The performance of the C4.5 algorithm differed only by 0.12% on both works. However, SVM performed better than C4.5 on Conway's work, while the reverse happened on the work by (Zhou *et al.*, 2005).

Next, Conway focused on the selection of feature sets. He divided the feature sets on the following groups: standard features (See (Sebastiani, 2002)), biographical features, syntactic features and keyword-based features. The standard features used were (i) the 2000 most frequent unigrams derived from the DNB, (ii) the 2000 most frequent unigrams with function words removed, derived from the DNB, (iii) the 2000 most frequent unigrams derived from the DNB stemmed using the Porter Stemmer, (iv) the 2000 most frequent bigrams derived from the DNB, (v) the 2000 most frequent trigrams derived from the DNB, and (vi) 319 function words [1].

The biographical features include: (i) pronouns (he, she, etc.) consisting in a boolean feature that gets fired if the sentence contains one or more pronouns, (ii) names, which defines six boolean features, namely, title (Mr., Ms., etc.), company (Microsoft, etc.), non commercial organization (army, parliament, etc.), forename (David, Dave, etc.), surname (Jackson, etc.), family relationship (mother, etc.), (iii) year, which is a boolean feature triggered if the sentence contains a year, and (iv) Date, which is a boolean feature triggered if there is a date in the sentence.

The choice of syntactic features was based on the data published as part of the research project described by Biber (Ferguson, 1992). Consequently, ten features were chosen, consisting of the five best and least characteristic of biography. The five most characteristic features were: (i) past tense, (ii) preposition (iii) noun, (iv) attributive adjective, (v) nominalization (nouns ending in tion, ment, ness, or ity). The five least characteristic features were: (i) Present Tense, (ii) Adverb (iii) Contraction, (iv) Second Person Pronouns, (v) First Person Pronouns. The referred features were identified using part-of-speech taggers, patterns or gazetteers. An alternative for selecting biographical features involves using key-keywords, exploiting the presence of common words on biographies, in order to identify biographical sentences. To discover the key-keywords, two related methods were used, namely naive key-keywords method and the WordSmith key-keywords method.

Moreover, Conway tested if the usage of "Bag-of-words" style sentence representation augmented with syntactic features provides a more effective sentence representation for biographical

[1] http://www.dcs.gla.ac.uk/idom/ir resources/linguistic utils/stop word

Fearure Set	Naive Bayes (%)	SVM (%)
DNB Unigrams	78,78	78,18
DNB Bigrams	69,08	71,28
DNB Trigrams	57,98	61,45
Syntactic Features	69,84	66,61
Syntactic Features and DNB Unigrams	80,68	77,72
Syntactic Features and DNB Bigrams	74,07	72,42
Syntactic Features and DNB Trigrams	64,54	69,30
Pseudo-Syntactic Features and DNB Unigrams	79,18	77,30

Table 2.6: Accuracy of Syntactic and Pseudo-syntactic Features (Conway, 2007)

sentence recognition than "bag-of-words" style alone. Both Santini and Stamatatos et al. concluded that syntactic features (noun phrases, verb, phrases, etc.) improved the accuracy of genre classification at the document level (Santini, 2004; Stamatatos *et al.*, 2000).

In the context of Conway's research, the pseudo-syntactic features are wording n-grams where $n > 1$. These research results are presented on Table 2.6 and show that both with Naive Bayes and SVM, unigrams performed better than bigrams and those performed better than trigrams.

However, in the previous experience (table 2.5), the Naive Bayes accuracy was almost 2% higher although the number of unigrams was decreased. Furthermore, these results disagreed with those obtained by Furnkranz, who claimed that trigrams yields the best results, and sequences greater than three reduce the classification accuracy (Furnkranz, 1998). Moreover, the previous comparison of Naive Bayes and SVM resulted in a superior performance of the Naive Bayes contradicting these new results. The results also showed that the use of syntactic or pseudo-syntactic in addition to the DNB unigrams, improved the results of Naive Bayes but lowered the results of the support vector machines. However, these results do not reach a significance level that allowed strong conclusions to be drawn (using the two tail corrected re-sampled t-test), which confirms the conclusions reported by (Santini, 2004; Stamatatos *et al.*, 2000), in which they claimed that there is a small accuracy gain in using syntactic features, although they did not refer is that gain is statistically significant. Another conclusion from this study is that the syntactic features performed better than the pseudo-syntactic features. After these experiments, Conway argued that it remains as an open question, whether this kind of small increase in accuracy can be gained for genre classification more generally or whether it is applied only to special case of biographical text classification (Conway, 2007).

Next, Conway explored whether the choice of frequent lexical items from a biographical corpus produces better accuracy for the biographical classification task than another lexeme based methods. For that purpose, he compared three alternative lexeme based methods with the 2000 most

Fearure Set	Naive Bayes (%)	SVM (%)
2000 DNB Unigrams (Baseline)	78,78	78,18
319 Function Words	75,43	73,59
2000 DNB Unigrams (stemmed)	79,93	78,92
1713 DNB Unigrams (No Function Words)	72,37	76,94

Table 2.7: Accuracy of Alternative Lexical Methods (Conway, 2007)

frequent unigrams in the DNB. The first alternative representation is based on the idea that function words can capture non-topical context of text. This feature representation was composed by 319 function words. The second alternative representation is based on stemming, which consist of deriving a word to its root form. The idea is that the canonical form will provide better classification accuracy for the biographical categorization task, because the key biographical words will be represented by a single feature. The third alternative representation contrast with the first, consisting on the removal of non-topical function words. The idea is that topic neutral function words are unlikely to contribute to classification accuracy. For this experiment, four feature sets were used, namely: (i) the 2000 most frequent unigrams from the DNB (baseline), (ii) 319 function words, (iii) the 2000 most frequent unigrams from the DNB in stemmed form, (iv) the 1713 most frequent unigrams from the DNB with function words removed. The obtained results are presented on Table 2.7.

The Table 2.7 shows that the stemmed DNB unigrams provided the best performance (79.93%), followed by the baseline (78.78%), but not at a statistical significant level. Moreover, both the usage of function words alone and the removal of existing function words caused a reduction on the classification accuracy at a statistical significant level compared to the unigram baseline. However, the differences between the accuracy scores are only 3.3%, which is not much if we consider that there are only 319 function words and 2000 frequent unigrams. It is also remarkable that with the naive bayes, the use of the “stop-worded” feature set (composed by the most frequent 2000 unigrams in the DNB minus the function words) performed worse than the function word feature set, despite the “stop-worded” feature set contained 1713 features, and the function word feature set only 319 features. Consequently, the results of (Holmes & Forsyth, 1995) were confirmed, which claimed that function words are important for genre classification.

Next, Conway explored two different related methods for selecting the genre specific key-keywords, namely naive key-keyword’s method and the WordSmith key-keywords method (Xiao & McEnery, 2005). The existing difference between the referred methods is that the naive method ranks keywords based on the number of occurrences of each word in the documents, contrarily to the WordSmith method which ranks keywords based on the number of documents in which the word

Fearure Set	Naive Bayes (%)	SVM (%)
500 Frequent Unigrams	81,25	76,07
500 Keywords	76,86	76,90
500 Naive Key-Keywords	78,92	78,32
500 WordSmith Key-keywords	68,34	63,11

Table 2.8: Accuracy of Keyword Methods (Conway, 2007)

Fearure Set	Naive Bayes (%)	SVM (%)
USC Feature	76,61	79,33
DNB/Chambers	76,58	79,32

Table 2.9: Classification Accuracies of the USC and DNB/Chambers Derived Features (Conway, 2007)

is a key. The research results are presented on Table 2.8.

The results showed that the use of key-keywords reduced the classification accuracy. However, the feature selection was performed using external data (Wikipedia and Chambers as a biographical corpus, and the BROWN corpus as a reference corpus), to avoid artificially inflating the classification accuracy. Thus, one can conclude that simple frequency counts provide better performance than the key-keyword methodologies.

Conway argued that these results were surprising, because the opposite result was expected. Moreover, he claimed that further research is needed before definitive conclusions can be drawn.

Before concluding his work, Conway tested the portability of feature sets, through a comparison of the best performing features identified by (Zhou *et al.*, 2005) (5062 unigram features from the USC biographical corpus) with a feature set with the same size consisting of frequent unigrams derived from a sample of the DNB and Chambers Dictionary of Biography.

Table 2.9 illustrates the experiment results, showing that the feature set performance was identical. These results were surprising, because some increase in classification accuracy using Zhou et al.'s labor intensive feature identification process was expected, when compared to a simple unigram frequency list derived from biographical dictionaries. However, (Zhou *et al.*, 2005) achieved very high classification accuracy with this method using USC test data. Even though, the results suggest that manual identification of appropriate biographical features does not bring benefits derived from biographical dictionaries, when applied to alternative biographical data.

2.3 Summary

This chapter focused three different types of systems capable of automatically extract biographical information, namely, Question Answering Systems, Summarization Systems and Extraction Systems. Several different techniques were detailed for each kind of system, although some techniques are used on more than one type of system. The analisys included a discussion of biographical text characteristics, comparison of features, taxonomies, classifiers, etc.. Unfortunately, independently of the system, extracting biographical information from textual documents still presents many challenges to the current state-of-the-art.

Chapter 3

Proposed Solution

This chapter describes the architecture of the proposed approach for extracting biographical information from Wikipedia documents written in Portuguese. The difference between this approach and the usual approaches on the information extraction literature is that, here, the sentence is the basic unit instead of tokens. The work reported in this chapter can be seen as an advancement over these previous approaches, in the sense that it explores the usage of a sequence labeling models, i.e., CRFs, as well as voting protocols, in the task of classifying sentences as belonging to biography-related categories.

3.1 Proposed Methodology

Figure 3.3 provides a general overview on the methodology proposed for extracting biographical facts from textual documents.

- The first steps concern with delimiting individual tokens over the documents, and also with

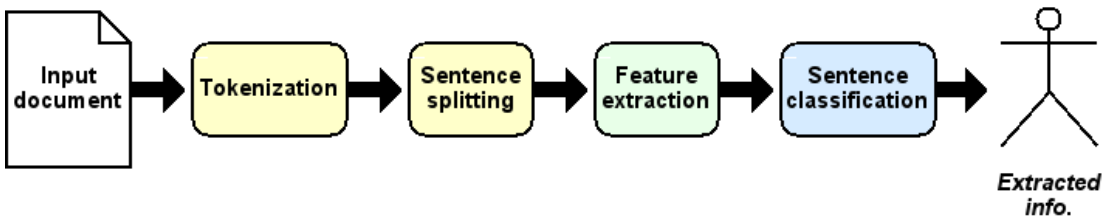

Figure 3.3: The proposed extraction method for biographical sentences

segmenting the documents into sentences. In our case, this is done through the heuristic methods implemented in the LingPipe[1] text mining framework.

- The third step concerns with the extraction of a set of features describing each of the sentences. The considered features are detailed in the subsection 3.5.
- The final step concerns classifying the sentences into one of the 19 classes mentioned in the section 3.2.

3.2 The Taxonomy of Biographical Classes

A biography can be defined as an account of the series of facts and events that make up a person's life. Different types of biographical facts include aspects related to immutable personal characteristics (i.e., date and place of birth, parenting information and date and place of death), mutable personal characteristics (i.e., education, occupation, residence and affiliation), relational personal characteristics (i.e., statements of involvement with other persons, including marital relationships, family relationships and indications of professional collaboration) and individual events (i.e., professional activities and personal events).

The above classes were taken into consideration when developing the information extraction approach described in this dissertation. It is also important to note that biographical facts can be expressed in the form of complete sentences, phrases or single words, although here we only model the problem at the level of sentences. Thus, our task of automatically extracting biographical facts essentially refers to classifying sentences into one of two base categories, namely biographical and non biographical. Furthermore, sentences categorized as biographical are sub-categorized as (i) immutable personal characteristics, (ii) mutable personal characteristics, (iii) relational personal characteristics, (iv) individual events, and (v) other. Sentences categorized as immutable personal characteristics are further sub-categorized as (i.1) date and place of birth, (i.2) parenting information, or (i.3) date and place of death. Sentences categorized as mutable personal characteristics are also further sub-categorized, in this case as either (ii.1) education, (ii.2) occupation, (ii.3) residence and (ii.4) affiliation. Sentences categorized as relational personal characteristics are further sub-categorized as (iii.1) marital relationship, (iii.2) family relationships, and (iii.3) professional collaborations. Finally, sentences categorized as individual events are sub-categorized as either (iv.1) professional activity, (iv.2) and personal events.

Figure 3.4 illustrates the hierarchy of classes that was considered. Although the presented categories are hierarchical in nature, in this work, we will handle them in two distinct ways:

[1] http://alias-i.com/lingpipe/

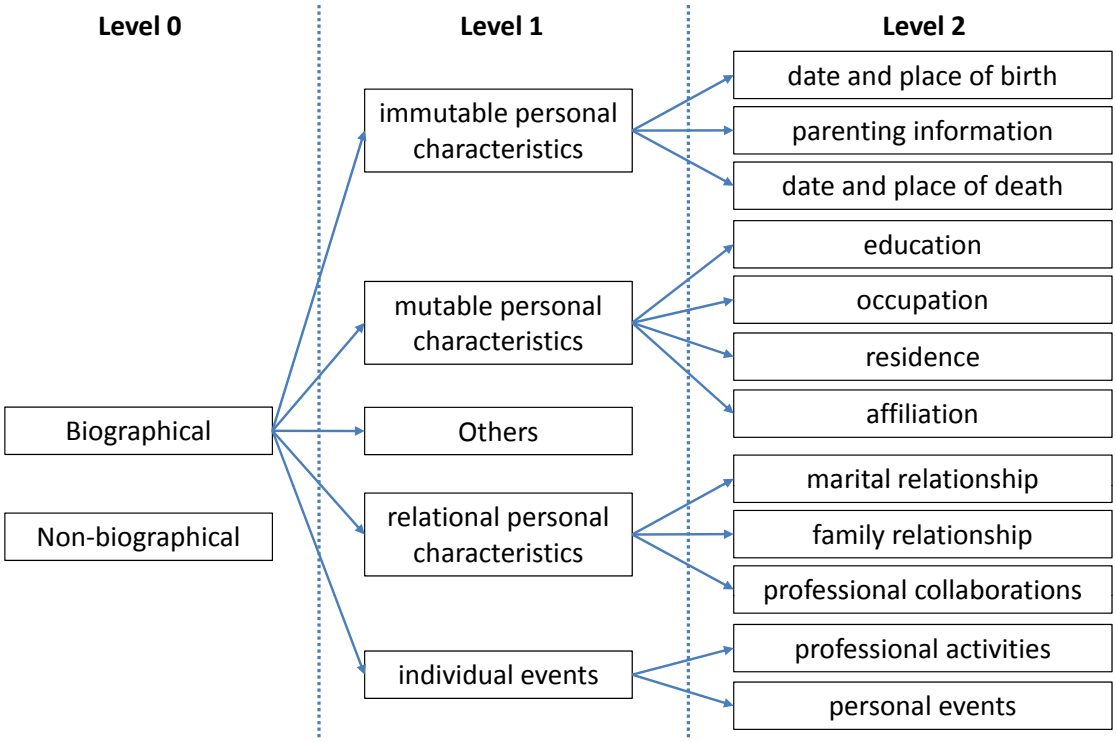

Figure 3.4: The hierarchy of classes considered in our tests.

- As a simple list of 19 different categories without any hierarchy levels. Thus, when performing the classification, each sentence receives the most specific label which covers all the sentence's topics. For instance, if a sentence contains birth information, it is tagged as *date and place of birth*, but if the sentence also has information about the person's death, then the sentence will be labeled as *immutable personal characteristics*, because it is the most specific label that covers all the sentence's topics. Similarly, if a sentence has information about a person's education and professional collaborations, it will be tagged with the *biographical* label.

- As the three-level hierarchy that is presented in the Figure 3.4. When performing the classification with this hierarchy, the classification is done independently for each level. Thus, first we consider the level 0 which contains only the label *biographical* and *non biographical*. Consequently, the remaining labels are reduced to one of their accepted ancestors at the considered level (e.g., *date and place of birth* are reduced to *biographical*). This technique is used for each of the three hierarchy levels, but notice that for each level, the previous level's labels are also considered (e.g., when classifying the level 1 of the hierarchy, the label *biographical* existent in level 0 is still valid and do not need to be reduced).

Notice that when working in the last level of the hierarchy, all the labels are valid in a way similar to what happened with the flat working hierarchy mode, since all labels are allowed. However, in the hierarchical mode each sentence receives a top-level label (*biographical* or *non biographical*), and will only receive a more specific label (from level 1 and later for level 2) if the required conditions are met, contrasting with the flat hierarchy mode in which all the labels are considered to be on the same level and with the same probability. Finally, several different methods to traverse the hierarchy or either choose or not a more specific label, are possible, and those will be described later on Section 4.1.

3.3 The Corpus of Biographical Documents

To build a corpus of gold-standard annotated data, we started by collecting a set of 100 Portuguese Wikipedia documents referring to football-related celebrities, like referees, players, coaches, etc. Afterwards, we performed the manual annotation of the documents, using the 19 different classes described in Section 3.2. Table 3.10 presents a statistical characterization of the resulting dataset. Recall that, each sentence received only one tag. Thus, the selected tag is the one that is most specific, and that covers all the sentence's topics.

3.4 Classification Approaches

Classification approaches such as Naive Bayes or Support Vector Machines assume that the sentences in each document are independent of each other. However, in our particular application, the biographical facts display a strong sequential nature. For example, a place of birth is usually followed by parenting information, and documents with biographies usually present facts in a chronological ordering, starting from information describing a person's origins, followed by his professional achievements, and so on. Based on these observations, we also experimented with a sequence labeling model based on Conditional Random Fields (CRFs), thus considering the inter-dependencies between sentences referring to biographical facts, in order to improve classification accuracy. The remaining of this section introduces the theory behind the NB, SVMs and CRFs models, since those are the most widelly used classifiers in the related literature.

	Value
Number of documents	100
Number of sentences	3408
Number of tokens	62968
Biographical	181
Immutable personal characteristics	4
Date or place of birth	2
Parenting information	20
Date or place of death	10
Mutable personal characteristics	4
Education	20
Occupation	212
Residence	6
Affiliation	5
Relational characteristics	1
Marital relationships	5
Family relationships	22
Professional collaborations	221
Individual events	78
Professional activities	484
Personal events	429
Others	221
Non biographical	1483

Table 3.10: Statistical characterization of the evaluation dataset.

3.4.1 Sentence classification with Naive Bayes

Naive Bayes (NB) is a probabilistic model extensively used in text classification tasks. Naive Bayes classifiers base their operation on a naive independence assumption, considering that each feature is independent of the others. Thus, using estimations derived from a training set, it is possible to perform the classification by calculating the probability of each sentence belonging to each class, based on the sentence's features, and then choosing the class with the highest probability. The probability of assigning a class C to a given sentence represented by features $F_1, \ldots, F_N$ is calculated using the following equation:

$$P(C|F_1, \ldots, F_N) = \frac{P(C)P(F_1, \ldots, F_N|C)}{P(F_1, \ldots, F_N)} \tag{3.1}$$

In text classification applications, we are only interested in the numerator of the fraction, since the denominator does no depend on C and, consequently, it is constant. See McCallum & Nigam (1998) for a more detailed discussion of Naive Bayes classifiers.

3.4.2 Sentence classification with Support Vector Machines

Support Vector Machines (SVM) are a very popular binary classification approach, based on the idea of constructing a hyperplane which separates the training instances belonging to each of two classes. SVMs maximize the separation margin between this hyperplane and the nearest training data points of any class. The larger the margin, the lower the generalization error of the classifier. SVMs can be used to classify both linearly and non-linearly separable data, through the usage of different kernel functions for mapping the original feature vector into a higher-dimensional feature space, they have been shown to outperform other popular classifiers such as neural networks, decision trees and K-nearest neighbor classifiers.

SVM classifiers can also be used in multi-class problems such as the one proposed in this dissertation, for instance, by using a one-versus-all scheme in which we use a number of different binary classifiers equaling the number of classes, each one trained to distinguish the examples in a single class from the examples in all remaining classes (Rifkin & Klautau, 2004). The reader should refer to the survey paper by Moguerza and Munoz for a more detailed description of SVMs classifiers (Moguerza & Muñoz, 2006). In this dissertation, we used an SVMs implementation relying on the one-versus-all scheme and on a radial-basis function as the kernel.

3.4.3 Sentence classification with Conditional Random Fields

The probabilistic model known as Conditional Random Fields offers an efficient and principled approach for addressing sequence classification problems such as the one presented in this dissertation (Lafferty *et al.*, 2001).

We used the implementation from the MinorThird toolkit of first-order chain conditional random fields (CRFs), which are essentially undirected probabilistic graphical models (i.e., Markov networks) in which vertexes represent random variables, and each edge represents a dependency between two variables.

A CRFs model is discriminatively-trained to maximize the conditional probability of a set of hidden classes $y =< y_1, \ldots, y_C >$ given a set of input sentences $x =< x_1, \ldots, x_C >$. This conditional distribution has the following form:

$$p_\Lambda(y|x) = \frac{1}{\sum_y \prod_{c=1}^{C} \phi_c(y_c, y_{c+1}, x; \Lambda)} \prod_{c=1}^{C} \phi_c(y_c, y_{c+1}, x; \Lambda) \tag{3.2}$$

In the equation, ϕ_c are potential functions parameterized by Λ. Assuming ϕ_c factorizes a log-linear combination of arbitrary features computed over the subsequence c, then $\phi_c(y_c, y_{c+1}, x; \Lambda) =$

$exp(\sum_k \lambda_k f_k(y_c, y_{c+1}, x))$ where f is a set of arbitrary feature functions over the input, each of which having an associate model parameter λ_k. The feature functions can informally be thought of as measurements on the input sequence that partially determine the likelihood of each possible value for y_c. The parameter k represents the number of considered features and the parameters $\Lambda = \{\lambda_k\}$ are a set of real-valued weights, typically estimated from labeled training data by maximizing the data likelihood function through stochastic gradient descent. Given a CRFs model, finding the most probable sequence of hidden classes given some observations can be made through the Viterbi algorithm, a form of dynamic programming applicable to sequence classification models.

The modeling flexibility of CRFs permits the feature functions to be complex, overlapping features of the input, without requiring additional assumptions on their inter-dependencies. The list of features considered in our experiments is detailed in the following section.

3.5 The Considered Features

In our experiments, we extracted various sets of features for training our classifiers. The considered features can be categorized as follows:

- Token features, including unigrams, bigrams and trigrams. The computation of these features starts by receiving a number N of n-grams to use, which was 500 in this case. The selected number of ngrams was chosen with the objective of not to be too big nor too small, since the analysis of the performance evolution with the increase of the number of ngrams was inconclusive.

  After listing all the unigrams, bigrams and trigrams from the training set, the application selects N of each. This selection is accomplished by removing repeatedly the most common and the most rare n-grams until the desired number of n-grams is obtained. Consequently, the most representative 500 n-grams are kept.

  The remaining n-grams will be used as binary features for each sentence, where the corresponding value is one if it appears in the sentence and zero otherwise.

- Token surface features, referring to the visual features observable directly from the tokens composing the sentence (e.g., whether there is a capitalized word in the middle of the sentence, whether there is an abbreviation in the sentence, whether there is a number in the sentence, or whether the sentence ends with a question mark or an exclamation mark). We use one binary feature for each of the previously listed properties.

- Length based features, corresponding to the number of words in a sentence. Each possible sentence length is used as a feature which gets fired for a sentence having that length.
- Position based features, corresponding to the position of the current sentence in the document. We include this feature based on the observation that the first sentences in a document usually contain information about the origins of a person, while subsequent sentences often contain information regarding the person's activities. Thus, each sentence is mapped to a value between 1 and 4 according to its position in the text which could be from the first until the forth quarter, respectively.
- Pattern features, consisting in a list of common biographical words. This list includes expressions such as *was born*, *died* or *married*. To implement this feature, a stemmer for the Portuguese language was delopped and used to stem the sentence's words. Then, each stemmed word is compared with the stem of our list of biographical words. If a match exists, then the feature value is 1, else it is 0.
- Named entity features, referring to the occurrence of named entities of particular types (e.g., persons, locations and temporal expressions) in the sentence. Each named entity type is mapped to a binary feature which gets fired for a sentence having at least one named entity of that particular type. The generation of these features uses a developed Portuguese named entity tagger that use the formalism of Hidden Markov Models and was trained using a modified version of a Portuguese dataset from Linguateca, known as HAREM[1]. The modifications include the remotion of all tags except the ones referring names, locations and time expressions, and also the selection of only one tag for each segment.
- Surrounding sentence features, referring to the features observable from the previous and following sentence in the text. Specifically, these features refer to the token, token surface, position, length, domain specific and named entity features computed from the two surrounding sentences.

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

3.6 Summary

This chapter presented the basis for a set of experiences that will be described in the next chapter, whose objective was to automatically extract biographical sentences. Thus, this chapter presented and detailed the classification models that will be compared, as well as the considered feature sets. Furthermore, the used corpus and the taxonomy of biographical classes were described. The next chapter will describe the experiments and discuss the results obtained.

Chapter 4

Evaluation Experiments

This chapter presents the experiments that were conducted in order to validate the proposed methods. First, it will be presented the evaluation methodology and the used metrics will be described. Then, the different experiments will be detailed as well as their motivation, followed by their respective results and a careful analisys of them.

4.1 Evaluation Methodology

Our validation methodology involved the comparison of the classification results produced with different combinations of feature groups and with different classifiers, against gold-standard annotations provided by humans. In addition, we have created two new classifiers referred as *Voting 1* and *Voting 2* which use voting protocols to make their decisions. Voting protocols consists of an unsupervised approach, where each voter ranks the available candidates, and the outcome is chosen according to some voting rule (Conitzer, 2006). The voting rule used in the context of this work is a variation of the rule known as *plurality rule*, in which each voter (NB, SVMs, CRFs) will vote in their preferred candidate, and the candidate with more votes is returned. Thus, both *Voting 1* and *Voting 2* use the tags returned by the voters (NB, SVMs, CRFs) to accomplish their own label assignment. The difference between *Voting 1* and *Voting 2* is that, when a voting draw occurs, the classifier *Voting 1* chooses the tag with biggest associated confidence, whereas the classifier *Voting 2* chooses the tag with more occurrences in the training data.

The used metrics were a macro-averaged precision, recall and F_1, weighted according to the proportion of examples of each class. The reason to use these metrics is the existence of large discrepancies between the number of examples available for each class. Thus, the use of these

metrics avoids the fact that the sentences with rare labels have a much higher weight than those with common labels.

The set of 100 hand-labeled documents was used for both training and validation, using five-fold cross validation for comparing the human-made annotations against those generated by the automatic tagger. In order to evaluate the impact of different feature sets, we divided the features presented in Section 3.5 into four groups, namely (i) token features and token surface features, (ii) length and position features, (iii) pattern and named entity features, and (iv) surrounding features.

Finally, several types of experiments were conducted, namely:

- **Flat Classification** - This experiment considers that all the possible tags of the hierarchy are on the same level, ignoring the hierarchy levels. Thus, one of the existing 19 possible labels is selected by the classifiers based only on the training data. The results of this experiment, and their analyses are presented in the section 4.3.
- **Three Level Hierarchy** - This experiment used the taxonomy hierarchy described in section 3.2 with all its three levels. Consequently, for each experiment, three measurements were made (one for each level). Recall that for the measurement of each level, only the existent labels on that level plus their ancestors are accepted, and the remaining ones must be reduced to one of its accepted ancestors (i.e., if we are evaluating the level 1, all the *personal events* labels must be reduced to the *individual events* label). Thus, it is possible to measure the results in three different granularity levels. Furthermore, different techniques exploiting the referred hierarchy were tried, namely:
  - **Branching Method** - There are two distinct ways to select a new tag based on the hierarchical branches:
    - **Fixed Branch** - In the fixed branch classification mode, when the level of the hierarchy change, it is impossible to give a new tag which is not a descendant from the tag given in the previous level.
    - **Non Fixed Branch** - Is the opposite to the fixed branch classification mode, in which the new suggested label does not need to be a descendant of the previous assigned label.
  - **Increase Detail** - There are two distinct ways to go deeper in the tag hierarchy:
    - **Biggest Confidence** - Choose the tag with the biggest classifier's confidence. In this mode, after been given a tag to a sentence at any given level, this tag will only be replaced at the next hierarchy level if the new tag has more confidence than the tag of the previous level.

* **Dynamic Threshold** - Choose the tag based on a dynamic threshold. With this method, the training data is used to determine the confidence threshold for each level and classifier, that should be used to accept new labels. Thus, even if a new label is suggested with bigger confidence than the previous label, it will only be accepted if the confidence of that classifier for the new label, is bigger than the computed dynamic threshold for that classifier in that level.

4.2 Summary of the Experimental Results

4.3 Experiments with Flat Classification

As already described, we tested separate models based on different feature combinations. The considered classifiers were Naive Bayes, Support Vector Machines, Conditional Random Fields, and also, *Voting 1* and *Voting 2* (hereby referred as VOT1 and VOT2), which base their classification on the previous classifiers' answers. This section has the objective of presenting and discuss the obtained results when using the flat classification scheme.

Figure 4.5 shows the F_1 results for the five referred classification models and for the different feature sets, as a weighted average of the results obtained for each of the five-folds.

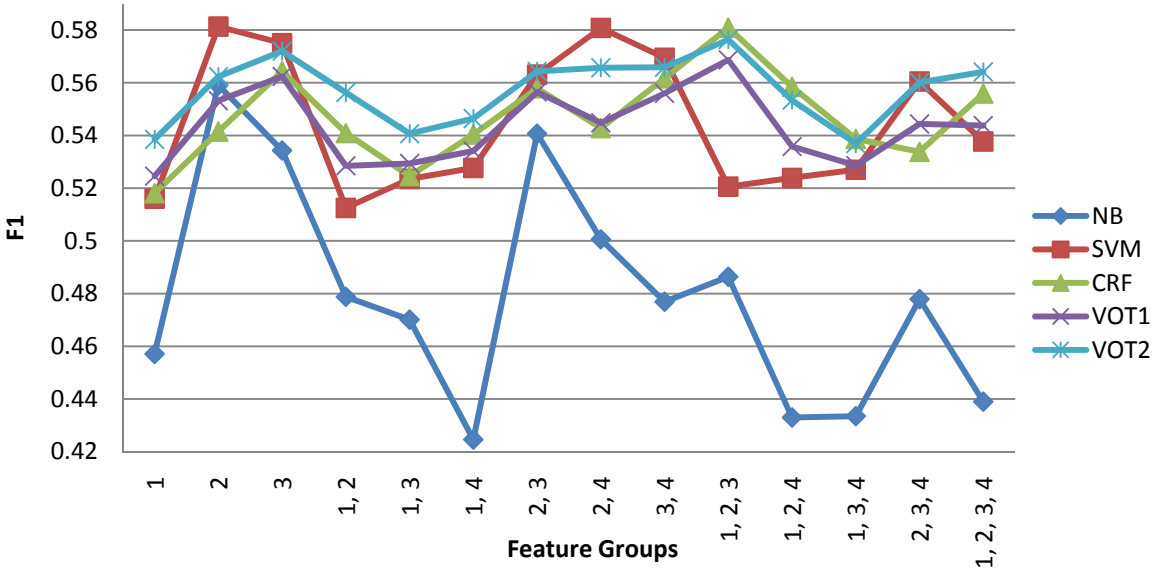

Figure 4.5: Comparison of classifiers using different features and a flat hierarchy

When comparing the results of all the classifiers, we can conclude that Naive Bayes had the worst results for almost every feature combination. Moreover, the performance of SVM and CRF was very similar, having a mean F_1 result of 54.42% and 54.72%, respectively. Furthermore, one can notice that the classifier VOT2 was always better than VOT1, also being the one with the best mean F_1 result (55.73%).

Surprisingly, the worst feature group was group (i), since the mean F_1 result of the described classifiers was only 51.08%. Moreover, all the obtained F_1 results bellow 44% included the feature group (i). The worst result was obtained when using only the features of the group (i) and its neighbours (group (iv)), obtaining a result of 42.46%. The highest classifier's mean F_1 results were obtained when the feature groups (ii) or/and (iii) were used. We also noticed that the feature group (iv) generally lowered the result of the remaining feature groups alone, except with CRFs.

Finally, the top three highest results included the SVMs with the feature group (ii) with an F_1 of 58.13%, the SVMs and group (ii) and (iv) with 58.08% and the CRFs with the features (i),(ii) and (iii), achieving and F_1 of 58.09%. However, the score difference between the top three results is almost unnoticeable.

Thus, the main conclusions are that NB was unsuitable for this type of classification, but the classifiers which use voting protocols are. Furthermore, we concluded that both SVMs and CRF had a similar performance. Relatively to the features used, we concluded that group (i) is not appropriate, but groups (ii) and (iii) are.

4.4 Experiments with Hierarchical Classifiers

The following experiments use the three-level hierarchy described in Section 3.2. This means that three distinct measurements were performed, one for each level, representing three levels of coarseness. Thus, before each experiment, all labels are reduced to an existent label on the measured level.

However, when a measurement is made, the results of the previous levels can be used to make new decisions. For example, when using the fixed branch hierarchy mode, the chosen label must be a descendant of the label chosen for that sentence, on the previous level. Moreover, in the experiments including the biggest confidence mode, a more specific label is chosen only if its confidence is bigger than the confidence of the label given on the previous level.

The two complementary modes include the non fixed branch hierarchy and the dynamic threshold mode. The non fixed branch hierarchy allows that a new label is assigned to a given sentence without being a descendant of the label given on the previous level. The dynamic threshold

ignores the confidence of the label given on the previous level, and tries to discover an optimum minimum confidence level in which it should accept a more specific label to a given sentence.

4.4.1 Experiments with Fixed Branch Hierarchy and Biggest Confidence

As already described, this experiment involves the classification of each sentence in three levels of coarseness, although the label attribution could stop at any hierarchical level. Moreover, for each level of coarseness, the new suggested label must be a descendant of the labels assigned for that sentence at the previous level, otherwise the new label will not be accepted.

The idea behind this experiment is that, as the hierarchy level increases, the number of available labels also increases, as well as the difficulty of assign a correct label. Consequently, the labels assigned in the previous levels should be taken into account when choosing the next label. Thus, in this experiment, we consider that the previous assigned label is correct, and consequently, we should only change it for another that is a subclass of the actual label, and also have a bigger confidence. The Figures 4.6 to 4.8 shows the F_1 results for the described experience.

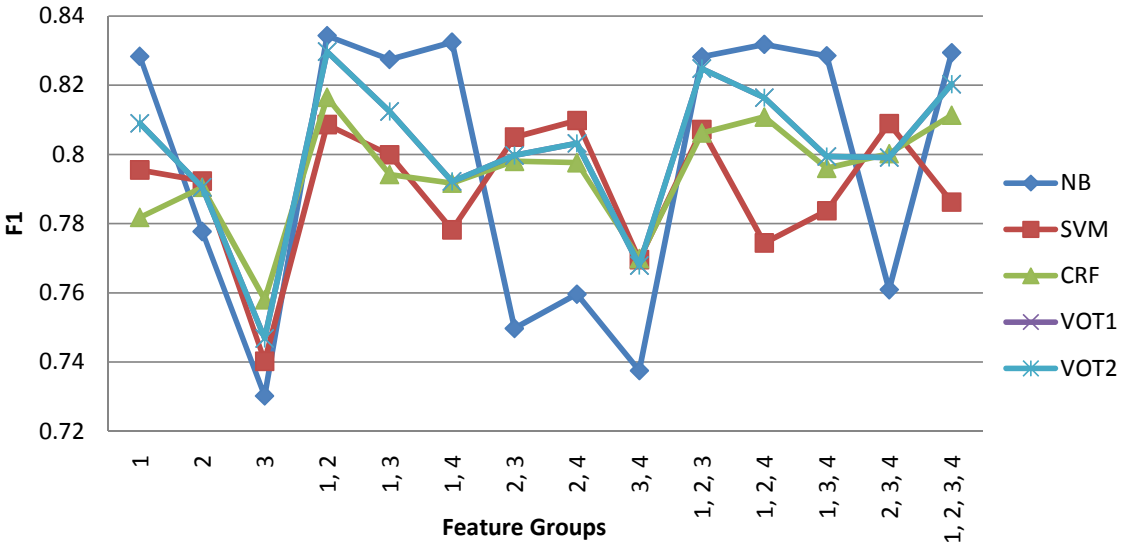

Figure 4.6: Level zero of the fixed branch hierarchy and biggest confidence experiment

Figure 4.6 shows the results for the level zero of the described experience. We could observe that both the classifiers VOT1 and VOT2 had the best mean performance (80.08%), obtaining exactly the same result for every combination of feature sets. The reason is that, in this experience, there are three voters (NB, SVMs and CRFs) and two possible labels, and consequently, a draw never occurs, which are the only case in which the behaviour of VOT1 and VOT2 differs. Furthermore,

both the voting classifiers obtained better results than SVMs and CRFs at almost every test. However, NB was the best-performing classifier for all feature combinations which contained the feature group (i) and the worst for any other combination. The worst performing feature set was group (iii) and the best was group (i), specially in conjunction with group (ii). The use of neighbour features (group (iv)) was inconclusive, since the results increased for some combinations and decreased for others.

Figure 4.7 shows the results for the level one of the described experience. We could observe that the classifier VOT2 had the best mean performance (59.62%) obtaining the top F_1 score for almost every feature combination. The second best mean performance belongs to the classifier VOT1, followed by CRFs, SVMs and finally NB. The best feature set was again the feature group (i) and the worst was group (iv), specially when used with NB. The best overall result was obtained with classifier VOT2 when using all the feature sets (63.10%).

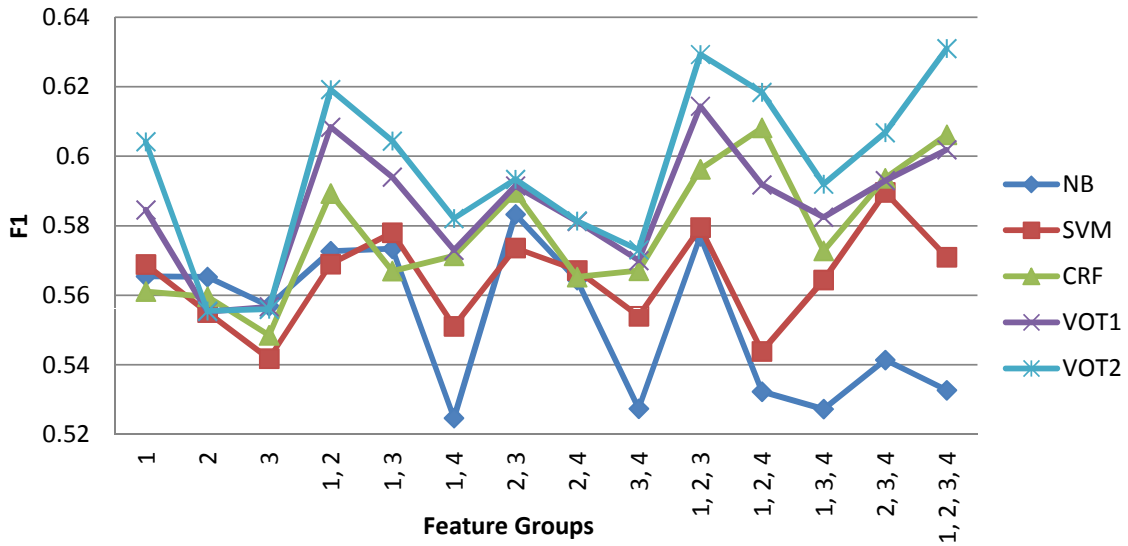

Figure 4.7: Level one of the fixed branch hierarchy and biggest confidence experiment

Figure 4.8 shows the results for the level two of the described experience. We could observe that again, the classifier VOT2 had the best mean performance (50.21%), obtaining the top F_1 score for almost every feature combination. The second best mean F_1 score belong to the classifier VOT1 (48.65%), followed by SVMs, CRFs and finally NB. The best feature set was group (i) and the worst was group (ii), although it helped to increase the results in conjunction with the group (i). The top F_1 score was obtained with the classifier VOT2 when using the feature groups (i), (ii) and (iii), obtaining an F_1 of 54.02%. Feature group (iv) had an unpredictable result one more time, since sometimes it helped to increase the performance, and sometimes it didn't, specially

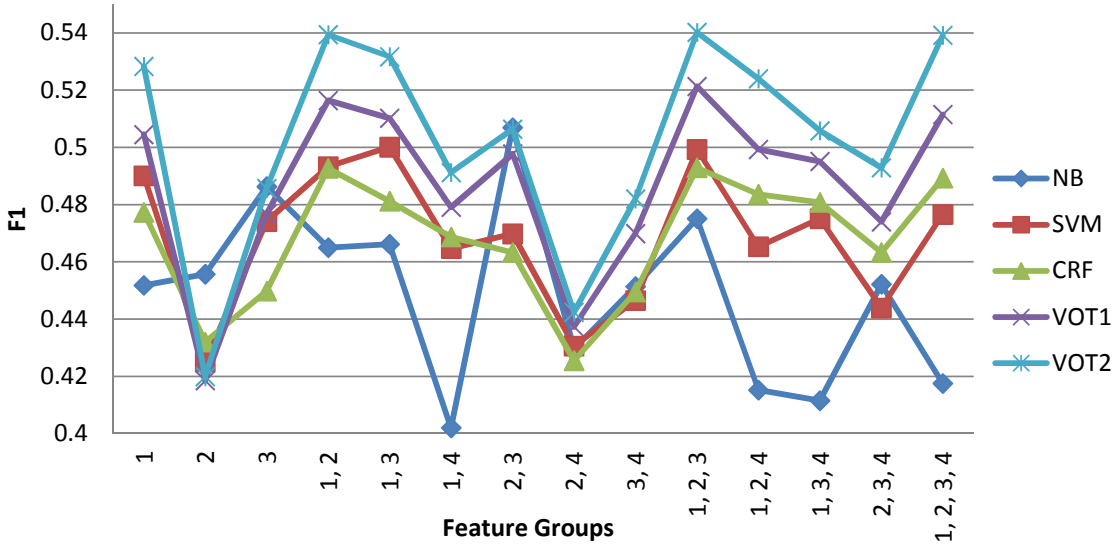

Figure 4.8: Level two of the fixed branch hierarchy and biggest confidence experiment

when using the NB classifier.

4.4.2 Experiments with Fixed Branch Hierarchy and Dynamic Threshold

The experiments reported on this section, are similar to those reported in Section 4.4.1, since the assignment of a new label requires the previous assigned label to be its ancestor. The difference is that the label replacement takes place if the classifier's confidence is bigger than a computed dynamic threshold, instead of simply being bigger than the confidence of the previous label.

The approach taken consists in the selection of a threshold value that minimizes the classification error on a validation set. In sum, for this experiment, the dataset was partitioned in three parts: training, validation and test set. The training set was used to train the classifiers, the validation set is used to discover the optimum threshold, and the test set is used to test the performance of the overall system.

In order to compute the threshold, the original training data was divided in two groups, namely, a train set composed by 80% of the documents and a validation set composed by the remaining 20%. Next, the classifiers were trained with the document's sentences existent in the newly created train set. Finally, the objective is to classify the document's sentences of the validation set, with all confidence values from 0% until 100%, and save the classification error for each different confidence level. Thus, if the classifier's confidence is less than the actual testing confidence level, then the answer is not considered, and it does not affect the classification error for that

confidence level. However, at least half of the sentences in the validation set must be classified in order to consider that confidence level as a candidate threshold. In the end, we know the percentage of right answers for each confidence level, and consequently, the confidence level that had the better results is selected as the threshold value. Then, both the training and validation sets are merged and used to train the classifiers in order to test them against the test set using the discovered threshold. The dynamic threshold is computed in each level for each classifier.

Figure 4.9, compares the results for the task of two label classification, known as level zero, for the described experience. We can observe that the classifiers based on voting protocols achieved the top mean F_1 score (80.24%), followed by NB (79.66%), CRFs (79.56%) and finally, SVMs (79.41%). However, the differences between the mean F_1 scores were almost unnoticeable.

We observed that the NB was the classifier whose behavior is more affected by the used features, achieving the best and the worst results. Furthermore, it is noticeable that the used features from the group (i) greatly affects the performance of NB. The combination of features of the group (i) with (ii) and/or (iii) also improved the performed of the other classifiers. However, the use of any of those groups of features alone resulted in a poor performance. The use of the feature group (iv) was inconclusive, although it helped to achieve the top result of this experience, which consisted on the use of NB and the features of groups (i), (ii) and (iv).

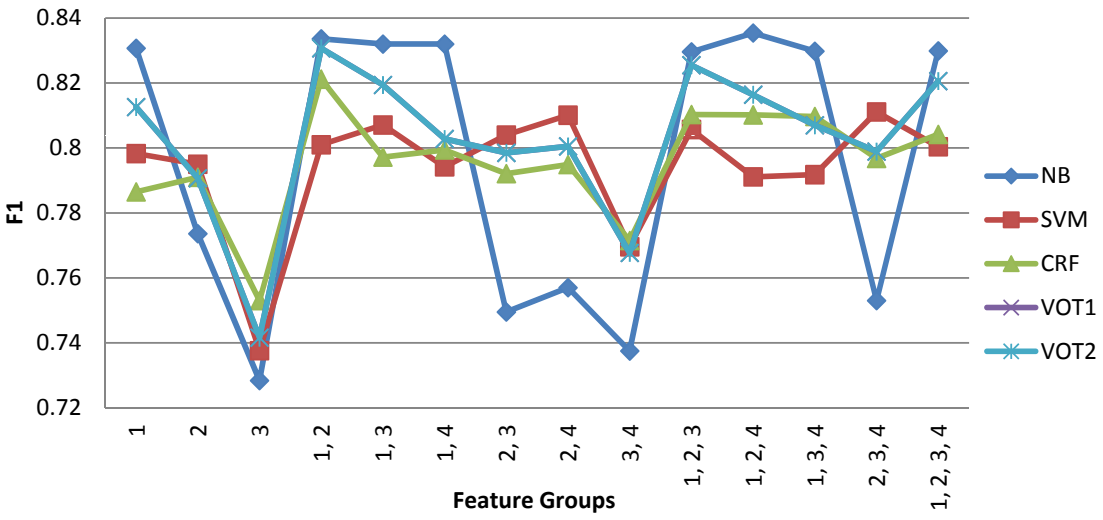

Figure 4.9: Level zero of the fixed branch hierarchy and dynamic threshold experiment

Figure 4.10, compares the results for the task of two label classification, known as level one, for the described experience. We can observe that the classifiers based on voting protocols achieved again the top mean F_1 scores, with VOT2 being the best (58.15%), followed by VOT1

(57.52%), SVMs (57.02%), CRFs (53.45%), and finally, NB (52.74%). However, one should have in attention that the CRFs did not return any answers when using the feature group (ii). Thus, if we did not consider the problematic feature, the mean F_1 results show that VOT2 achieved again the best performance (58.51%), followed by VOT1 (57.86%), CRFs (57.56%), SVMs (57.14%), and finally, NB (52.64%). Although the voting classifiers had achieved the top score with almost every combination on feature groups, it is interesting to note that when more than two groups of features are combined, the performance of the CRFs approaches or even surpasses the performance obtained by the classifier VOT2. Also noticeable is the poor performance of the NB with almost every combination of feature groups.

Relatively to the use of different groups of features, we concluded that, even though the use of groups (ii) or (iii) alone achieved poor results, when combined, they produce the best results. Moreover, although the use of the feature group (i) alone produce good results, its performance does not increase when combined with group (ii) or (iii). Finally, we concluded that the best individual result was obtained by VOT2 when using the feature groups (i), (ii) and (iii) or also adding the group (iv) achieving an F_1 score of 60.84%.

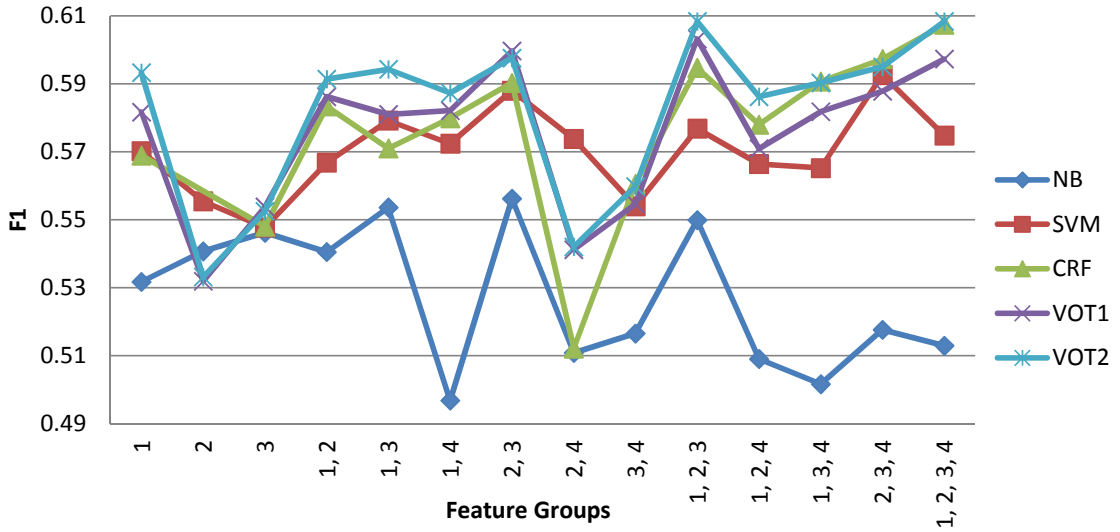

Figure 4.10: Level one of the fixed branch hierarchy and dynamic threshold experiment

Figure 4.11, compares the results for the task of two label classification, known as level two, for the described experience. Once again, the classifiers based on the voting protocols achieved the best mean F_1 scores, with VOT2 being the best (50.48%), followed by VOT1 (48.55%), CRFs (47.51%), and finally, SVMs (47.15%). Surprisingly, NB surpassed the performance of SVMs, although it was the by far the worst classifier for several feature combinations. Furthermore, we

noticed that the performance of SVMs and CRFs were identical in the majority of the tests.

We also observed that the feature group (i) alone produced good results, which increased a little when combined with group (ii) and (iii), which achieved the best individual result with the classifier VOT2. Moreover, the feature (iv) seemed to affect negatively the performance of the classifiers, specially NB, and SVMs.

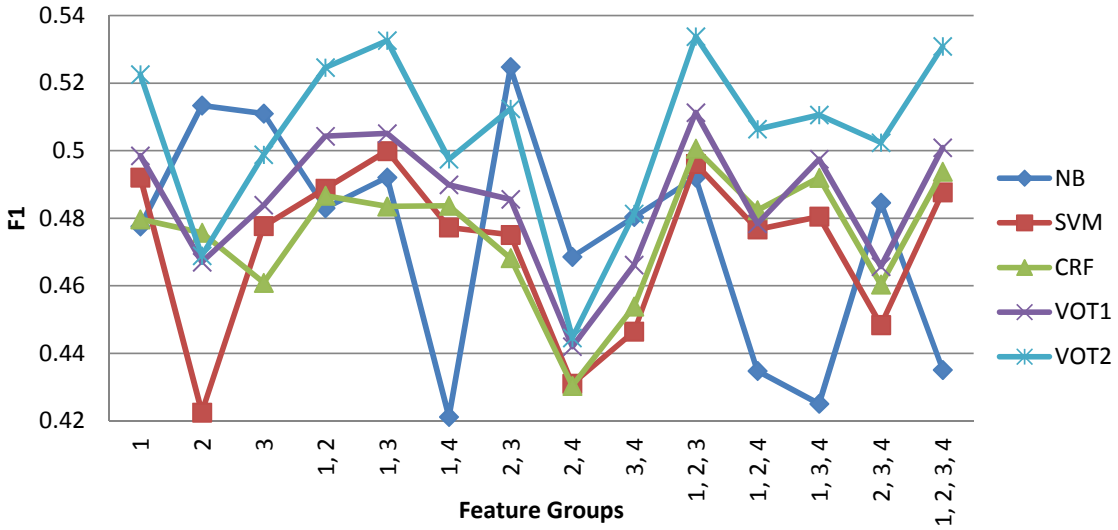

Figure 4.11: Level two of the fixed branch hierarchy and dynamic threshold experiment

4.4.3 Experiments with Non Fixed Branch and Biggest Confidence

This experiment is similar to the one described in the section 4.4.1, with the difference that now the new suggested label does not need to be a descendant of the previous assigned label. Although one can think that in this way, the work realized in the previous hierarchy level is lost, in fact, this is not true, since the new label must have a bigger confidence than the confidence of the previous label. Moreover, if the classifier miss assigned a label in any of the more coarse levels, now there is a chance to correct it in the next levels.

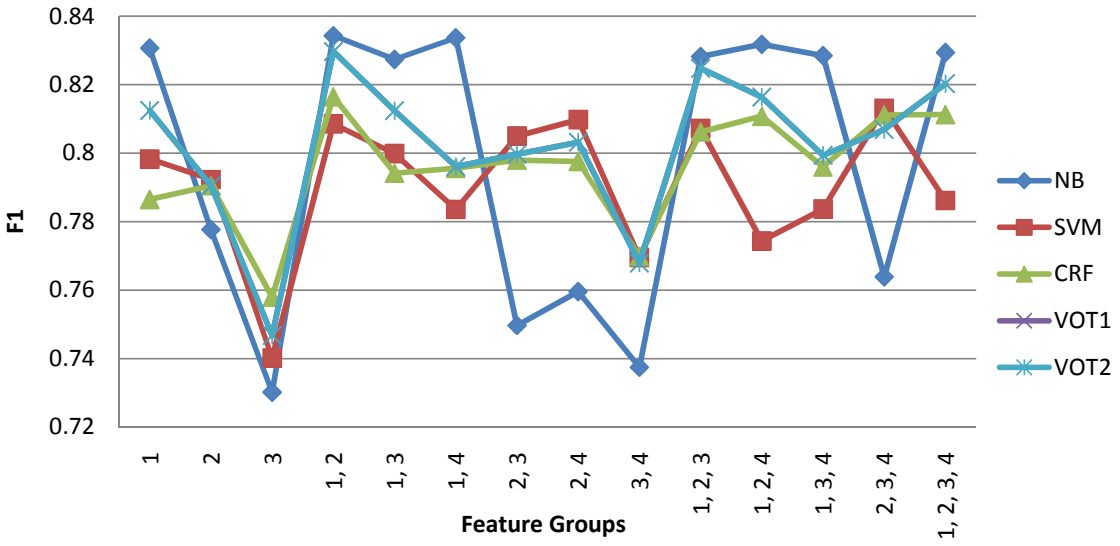

Figure 4.12: Level zero of the non fixed branch hierarchy and biggest confidence experiment

Figure 4.12, compares the results for the task of two label classification, known as level zero. We could conclude that NB had the best and the worst result in the test (83.43% and 73.02% respectively). However, the higher mean F_1 score belongs to the voting classifiers, which had exactly the same performance due to the reasons referred in section 4.4.1. The next best classifier was NB, followed by CRF and finally by SVM. However, SVM and CRF had a very similar performance, 79.59% and 79.09% respectively.

We also observed that the best feature set in this test was group (i), specially noticeable when using NB, since the worst F_1 result was 82.74% which is less than 1% below the maximum registered on this experiment. Notice also that the best F_1 score of NB without the use of features in the group (i) is 76.39% and the worst F_1 score when using feature group (i) is 82.74%. Moreover, we concluded that the worst performance feature group was the group (ii), followed by group (iii), although those helped to increase the results obtained with the group (i). We could not take any

conclusions from the use of the feature group (iv), since sometimes the results increased and in the others decreased.

Figure 4.13 compares the results for the described experiment for the level one of the hierarchy. The NB classifier was by far the worst classifier, since it obtained the worst results for every feature combination. The best classifier was again the VOT2 with a mean F_1 of 62.52%, followed by CRF with 62.33%, VOT1 with 61.65%, SVMs with 60.31%, and finally NB with 56.55%. Relatively to the features, a radical change occurred when comparing with the results of the previous level, since the best-performing feature set was group (ii) and even the group (iii) performed better than the group (i). One more time, the behaviour of the neighbour features was inconclusive.

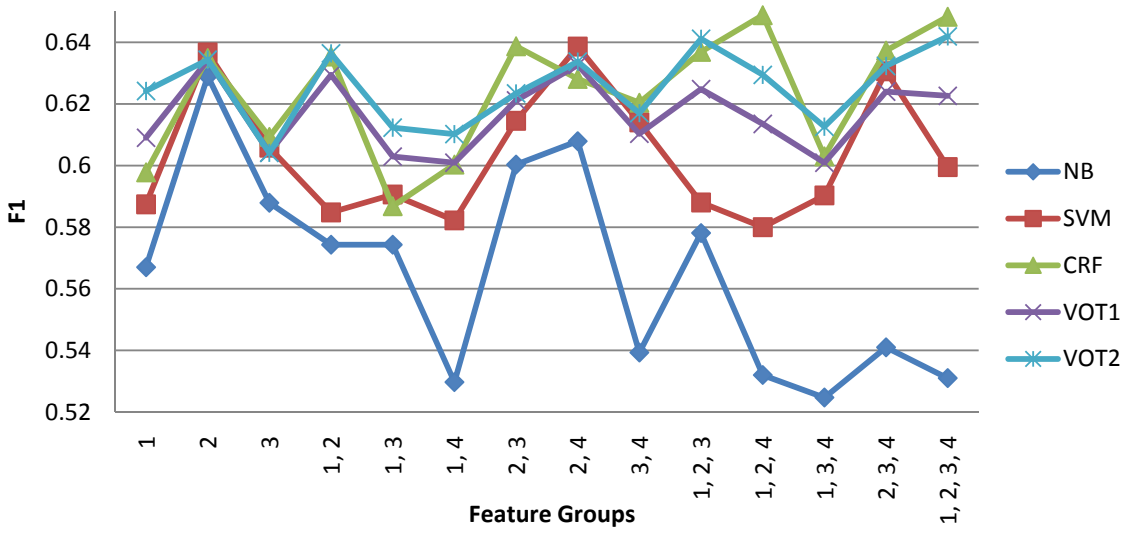

Figure 4.13: Level one of the non fixed branch hierarchy and biggest confidence experiment

Figure 4.14 compares the results for the described experiment for the level two of the hierarchy. The conclusions for this level of coarseness are remarkably similar to those of the previous level. Again, classifier VOT2 has the best mean F_1 score of 55.92%, followed by the CRF with 55.08%, VOT1 with 54.55%, SVMs with 53.79% and finally, the NB with 57.81%, which had the worst results in all the combinations of feature sets. Furthermore, the feature group (ii) was again the best-performing feature, and the group (i) was the worst, specially when combined with the feature group (iv). The best result was obtained with the SVMs and the feature group (ii), resulting in an F_1 of 58.73%.

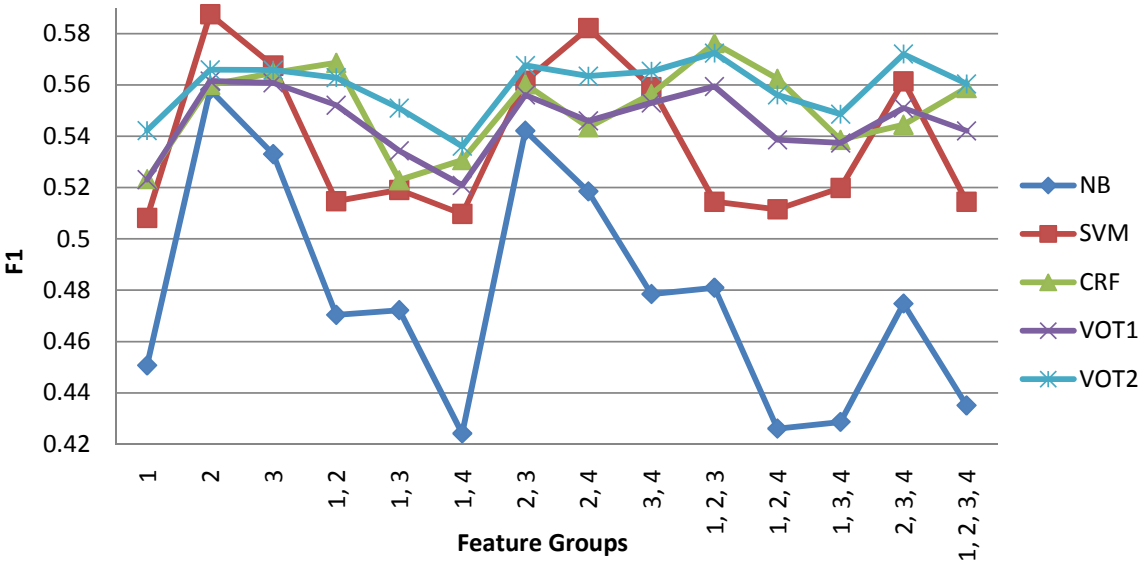

Figure 4.14: Level two of the non fixed branch hierarchy and biggest confidence experiment

4.4.4 Experiments with Non Fixed Branch and Dynamic Threshold

This experiment follows the same principle than the previous one (Section 4.4.3), with the difference that, now the new suggested label must have an associated confidence bigger than a computed dynamic threshold, instead of the previous label's confidence. As explained in section 4.4.2, the idea is to use the information presented on the training data to find an optimum threshold which should maximise the classification results, instead of relying only on the previous label's confidence.

Figure 4.15, compares the results for the task of two label classification, known as level zero, for the described experience. We can observe that the classifiers which used the voting protocols had the best mean F_1 performance (80.25%), followed by NB (79.68%), CRFs (79.54%), and finally by SVMs (79.25%). However, notice that in this experiment, the voting classifiers achieved the best result with only one feature combination (with group (i), (ii) and (iii)). Thus, we can conclude that the voting classifiers weren't recommended since for almost every feature combination another classifier can have a better performance, although they have the best mean performance. Furthermore, we noticed that the CRFs and SVMs had a very similar performance, and the NB was again capable of the best and worst results.

A careful analysis of the features showed that the combination of group (i) and (ii) yield the best mean F_1 results. Furthermore, the feature group (iii) achieved the worst mean F_1 score, although it had improved the results of feature groups (i) and (ii) achieving the top mean F_1 result.

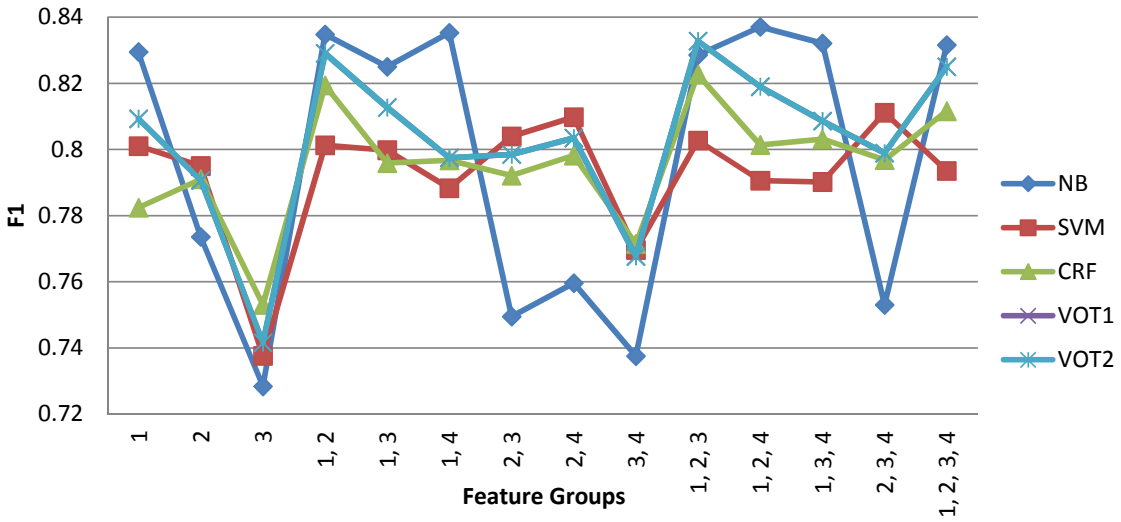

Figure 4.15: Level zero of the non fixed branch hierarchy and dynamic threshold experiment

The use of the feature group (iv) was inconclusive. The best result on this experiment was achieved by NB when using the feature groups (i), (ii) and (iv), with an F_1 score of 83.71%.

Figure 4.16 compares the results for the described experiment for the level one of the hierarchy. In this experiment, we observed that the best mean F_1 result was obtained by the classifier VOT2 (60.83%), followed by SVMs (60.42%), VOT (60.02%), SVMs (52.74%) and finally, NB (52.63%). However, one should notice that the CRFs did not return any result for the feature groups (ii) and feature groups (ii) with (iv). Thus, if we did not consider the problematic feature combination, the best classifier is again VOT2 (61.90%), followed by CRFs (61.53%), VOT (60.97%), SVMs (59.89%) and finally, NB (52.68%). Notice that NB got the worst result with almost every feature combination (except with group (ii)).

A careful analysis of the features used in this experiment revealed that the use of the feature groups (ii) and (iii) yielded the best mean F_1 (61.48%). The use of the feature group (iv) was again inconclusive. The best F_1 score in this experiment was achieved by CRFs when using the feature groups (i), (ii) and (iii).

At last, Figure 4.17 compares the results for the described experiment for the level two of the hierarchy. One more time, the classifier VOT2 achieved the best mean F_1 result (56.27%), obtaining the top result for almost every feature combination. The second best mean F_1 score belong to the classifier VOT1 (54.15%), followed by the SVMs (53.49%), CRFs (50.98%) and finally, NB (47.37%). However, notice that the CRFs did not return any result when used with the feature groups (ii) and (iv). Thus, if once more we exclude the problematic features, the best classifier

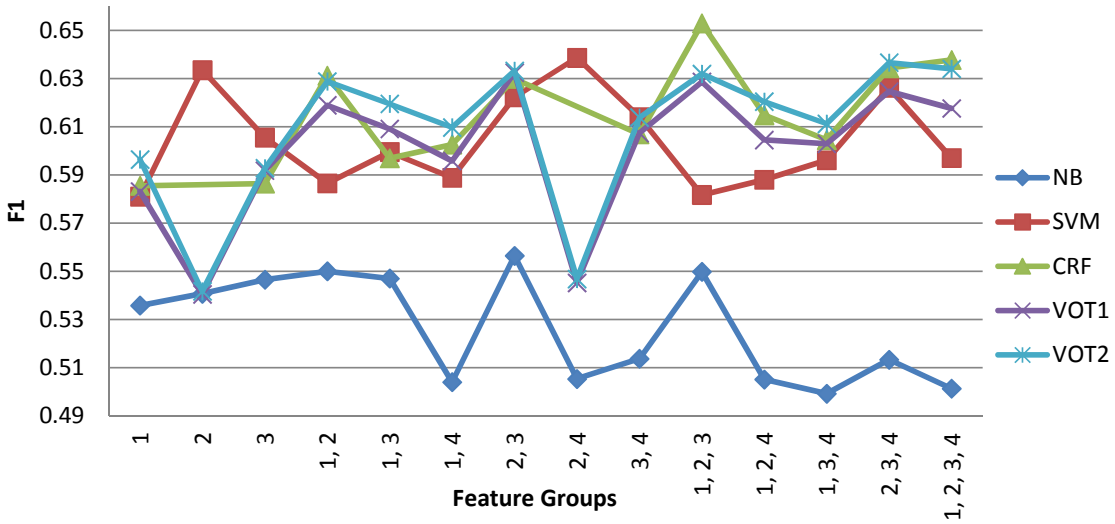

Figure 4.16: Level one of the non fixed branch hierarchy and dynamic threshold experiment

is VOT2 (56.48%), followed by the CRFs (54.90%), VOT1 (54.49%), SVMs (53.13%), and finally, NB (47.35%). Similarly to what happened on the previous level, NB obtained the worst results for almost every feature combination except with the group (ii). The observation of the performance of the features revealed several similarities between this level and the previous one. Thus, the feature groups (ii) and (iii) obtained the top mean F_1 result, and the usage was the group (iv) was one more time inconclusive. We also observed that, in general, the use of the feature group (i) lowered the results of the feature groups (ii) or (iii). Finally, we concluded that the best result on this experience was obtained by the classifier VOT2 when using the feature groups (ii), (iii) and (iv) with an F_1 score of 58.74%. However, several similar results exist on this experience.

4.5 Classification Results for Individual Classes

This section focus on the precision, recall and F_1 scores of each individual class for the top-performing combinations on each hierarchy level.

Table 4.11 shows the individual class results of the top-performing combination of feature, modes, and classifiers, for the hierarchy level zero. Those results were obtained with the NB classifier using the feature groups (i), (ii) and (iv), with the non fixed branch mode and dynamic threshold, which resulted in a global F_1 score of 83.71%

Table 4.12 shows the individual class results of the top-performing combination of feature, modes, and classifiers, for the hierarchy level one. Those results were obtained with the CRFs classifier

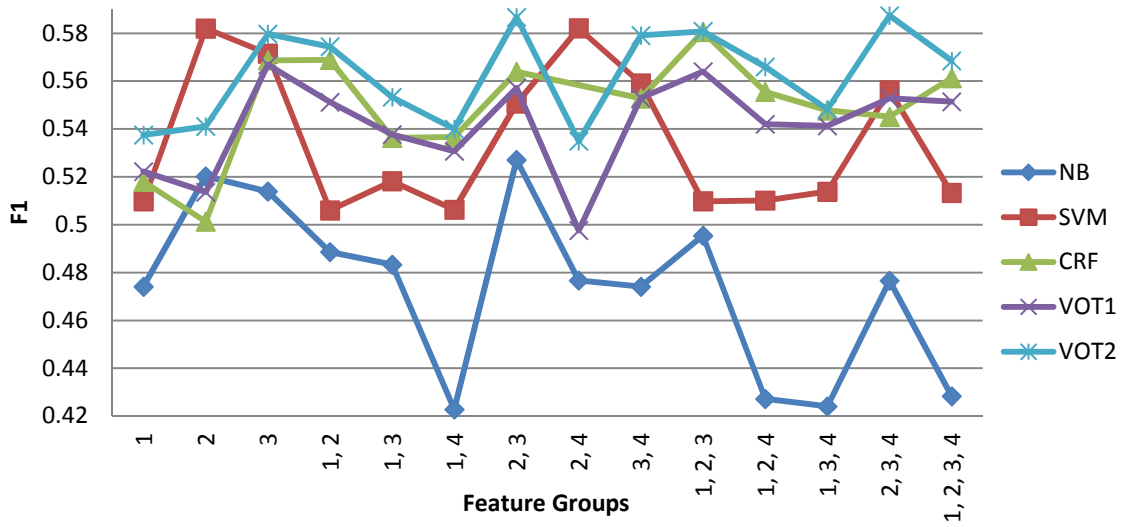

Figure 4.17: Level two of the non fixed branch hierarchy and dynamic threshold experiment

	Precision	Recall	F_1
Biographical	0.7637	0.9652	0.8527
Non biographical	0.9312	0.6123	0.7388

Table 4.11: The datailed class results of the best combination for the hierarchy level zero

using the feature groups (i), (ii) and (iii), with the non fixed branch mode and dynamic threshold, which achieved in a global F_1 score of 65.29%. Notice that the label *Immutable Characteristics* was omitted because its result was zero. The reason for that can be justified by the fact that it was the class with fewest training examples, since there is a correlation between the number of training examples and the corresponding results.

Table 4.13 shows the results obtained with a VOT2 classifier using the features from group (ii), (iii) and (iv), for each of the individual classes.

	Precision	Recall	F_1
Biographical	0.5155	0.4586	0.4854
Mutable characteristics	0.4348	0.3644	0.3965
Relational characteristics	0.2973	0.0884	0.1363
Individual events	0.5588	0.6327	0.5935
Other	0.3282	0.1946	0.2443
Non biographical	0.7326	0.8463	0.7854

Table 4.12: The datailed class results of the best combination for the hierarchy level one

	Precision	Recall	F_1
Biographical	0.3846	0.4696	0.4229
Immutable characteristics	0.2500	0.2500	0.2500
Date and place of death	1.0000	0.4000	0.5714
Occupation information	0.5000	0.1462	0.2262
Family relationships	0.5000	0.1364	0.2143
Professional relationships	0.3750	0.0136	0.0262
Professional activities	0.2865	0.3079	0.2968
Personal events	0.3090	0.1282	0.1812
Other	0.3333	0.0498	0.0867
Non biographical	0.5687	0.9090	0.6997

Table 4.13: The datailed class results of the best combination for the hierarchy level two

Notice that the classes whose result was zero were omitted. Furthermore, we observed that the missing classes were the ones with fewer examples on the dataset. Thus, we believe that the global results were seriously affected by the lack of training examples.

4.6 Summary

From the presented experiments, we could conclude that the Voting classifiers are appropriated for the task of biographical sentence extraction, specially the one that in case of a draw occurrence, chooses the tag with more occurrences in the training data (referred as VOT2). Moreover, we concluded that both CRFs and SVMs had a similar performance, although CRFs is generally better. On the other hand, NB was capable of the top and worst results, since his behaviour is very sensitive to the features used. Thus, NB could be very useful for the task of biographical sentence classification if a good selection of features was undertaken.

Relatively to the use of features, we concluded that the number of features used in an experience does not influence the performance obtained. Thus, it is preferable to use few carefully chosen than a lot of them chosen at random.

Although there is not a clear winner feature group, and the results greatly vary between experiments, we have noticed some patterns:

- When considering the finest coarseness level, the performance of NB is negatively affected when the feature group (iv) is used, specially in conjunction with the feature group (i). However, the opposite is true when considering the coarsest level of the hierarchy.

- For the NB classifier, the feature group (ii) and (iii) generally produced the best results when

considering the most finest coarseness levels, but was one of the worst feature combinations when considering the most coarse hierarchy level.

- The combination of feature groups (i) and (iii) produced the best results for the fixed hierarchy experiments when considering the SVMs. However, the reverse occurred in the non fixed hierarchy, in which the group (ii) or the group (ii) and (iv) was the best choice.
- For the SVMs classifier, the feature group (iii) was always the worst choice, when considering the most coarseness level, while the group (ii) was a good choice.
- For the CRFs classifier, the combination of feature groups (i) and (ii) produced the top results in all the experiments, specially when combined with the group (iii) which yielded the best result in the finest grained level of the hierarchy.
- The worst feature group for the CRFs classifier was the group (iii) for the most coarse level of the hierarchy, and the group (ii) or group (iii) or the combination of groups (ii) and (iv) for the hierarchy levels one and two.
- Both the voting classifiers had very similar results, which was expected since its behaviors differed only in case of voting draws.
- For the classifiers based on the voting protocols, the combination of feature groups (i) and (ii) generally achieved good results, specially when combined with the group (iii) which yielded the best results for all hierarchy levels.
- The worst feature group for the voting classifiers was the group (iii) for the most coarse level of the hierarchy, and the feature groups (ii) or group (ii) and (iv) for the hierarchy level one and two (except for the non fixed branch and biggest confidence experiment).

Relatively to the tests which use the biggest confidence mode and the ones which use dynamic threshold mode, we noticed that the dynamic threshold is best for the finest coarse level (hierarchy level 2), but the biggest confidence mode is best for the experiments on the hierarchy level one. Moreover, when comparing the experiments which involved the fixed branch and the ones which used the non fixed branch mode, we observed that the results for the non fixed branch experiments achieved far better results. Finally, when comparing the results of the flat hierarchy experiment with the non fixed branch hierarchy and dynamic threshold experiment, we noticed that both had a similar performance, and the best one depends on the used feature groups.

The top individual results for each hierarchy level were:

- For the hierarchy level zero the best F_1 score was 83.71% when using the NB classifier with the feature groups (i), (ii) and (iv) with the non fixed branch hierarchy and dynamic threshold mode.
- For the hierarchy level one the best F_1 score was 65.29% when using the CRFs classifier with the feature groups (i), (ii) and (iii) with the non fixed branch hierarchy and dynamic threshold mode.
- For the hierarchy level two the best F_1 score was 58.74% when using the VOT2 classifier with the feature groups (ii), (iii) and (iv) with the non fixed branch hierarchy and dynamic threshold mode.

Despite the relatively good overall results, when analyzing the classification results for each individual class, we noticed that the number of classes with a zero F_1 score increased with the increase of the hierarchy level. Furthermore, there is a direct correlation between the number of examples in the training set for a given type of sentence, and the number of correct answers for that same type. Thus, we believe that the presented results could be improved if the available dataset was increased, specially for the finest hierarchy level, in which nine classes obtained a zero F_1 score.

Chapter 5

Conclusions and Future Work

This dissertation, addressed the task of automatically extracting meaningful biographical facts, specifically immutable (e.g., place of birth) and mutable (e.g., occupation) personal characteristics, relations towards other persons and personal events, from textual documents published on the Web. We approached the task as a sentence classification problem, and we experimented with classification models based on the formalisms of Naive Bayes (NB), Support Vector Machines (SVMs), Conditional Random Fields (CRFs), and two other classification methods, which used voting protocols, also using various sets of features for describing the sentences.

Distinct forms of classification were attempted also considering three different levels of coarseness, forming a hierarchy. Those forms of classification varied in the method used to traverse the hierarchy, and the method used to decide to choose or not, a finer grained class. From the experiments, we concluded that the non fixed hierarchy branch mode, and the dynamic threshold mode produced the best results. However, the simplest classification experiment which used the flat hierarchy produced similar results.

Although many differences exist between this work and the Zhou et al. and Conway's work, we noticed that the best results, in the level zero of the hierarchy (biographical vs non biographical), was remarkably similar (83.71%, 82.42% and 80.66%), also confirming the results reported by other authors (Lewis, 1992).

We also observed that the overall F_1 results decreased with the increase of the hierarchy level, which was expected since the number of possible classes increases as well. However, despite the good overall results, some individual classes obtained a zero F_1 score, due to the reduced number of examples, specially on the finer hierarchy level. Thus, we believe that the presented results could be improved if the available dataset was increased, specially for the finest hierarchy

level, in which nine classes obtained a zero F_1 score.

The experiments allowed us to conclude that the classifiers based on voting protocols obtained generally the best results. However, there is not a clear classifier winner, since the results greatly varied from experience to experience. Similarly, there is not a winner feature combination. Consequently, the best combination will always depend on the objective of the classification. Furthermore, one can be interested on using a given combination to classify a given type of sentence even though that same combination produces bad overall results.

We think that classifying biographical sentences from textual documents still presents many challenges to the current state-of-the-art, specially when considering more than two classes. Thus, the returned results can hardly be used by other applications. Moreover, we believe that the existence of a common taxonomy and validation corpus would be extremely useful in order to compare different solutions, since actually the only thing in common between the most important works is the first level of the hierarchy (biographical and non biographic). Finally, it would be interesting to verify if the reported tendencies are portable to other areas not restricted to biographical information extraction and vice-versa, in order to incorporate new ideas in the field of biographical information extraction, from other information extraction areas.

5.1 Contributions

The main contributions of this work are described in the following subsections.

5.1.1 Biographical Taxonomy

In the context of biographical information extraction, from articles focusing football-related personalities, several documents were analyzed in order to produce a taxonomy that could be used to classify the majority of the existent sentences. The resulting taxonomy was composed by nineteen different classes, which could be grouped in three hierarchical levels.

- Level zero: Biographical, non biographical.
- Level one: Immutable personal characteristics, mutable personal characteristics, relational personal characteristics, individual events, others.
- Level two: Date and place of birth, parenting information, date and place of death, education, occupation, residence, affiliation, marital relationship, family relationship, professional collaborations, professional activities, personal events.

The hierarchical relationship of the referred classes is described in the section 3.2. The reason that motivated the creation of the presented taxonomy instead of adopting an existing one, was related with the specificity of the theme (football-related celebrities) that does not allow an easy sentence classification with the existent taxonomies.

5.1.2 Creation of a Portuguese Corpus

In order to apply the proposed methods, a Portuguese corpus was created based on Portuguese written Wikipedia's documents. We started by collecting a set of 100 Portuguese Wikipedia documents referring to football-related celebrities, like referees, players, coaches, etc. Afterwards, we performed the manual annotation of the documents, using the nineteen different classes described in Section 3.2. Notice that, each sentence had a unique sentence assigned. Thus, the selected tag is the one that is most specific, and that covers all the sentence's topics.

5.1.3 Comparison of classification methods

In order to classify the sentences, several classifiers were compared, as well as the different combination of features, decision methods and also in different granularity levels. The considered classification models were the Naive Bayes, Support Vector Machines, Conditional Random Fields and two other classifiers, which base their behavior in a voting protocol. Different feature set was also compared, namely, token features, token surface features, lenght based features, position based features, pattern features, named entity features and surrounding sentence features. The different classification methods included the method used to traverse the hierarchy and how to decide if the new suggested class is accepted or not. All the referred variable's combinations were compared for each of the three hierarchical levels of the taxonomy. The results were presented, analysed and the observed tendencies were highlighted.

5.1.4 Development of a prototype

The developed prototype used to accomplish the related experiments was made available online as an open-source package on Google Code, acessible through the following website: `https://code.google.com/p/biographicalsentenceclassifier/`.

5.1.5 Publication of the results

Some results reported on this dissertation were also partially published as an article in the proceedings of the 15th Portuguese conference on Artificial Intelligence known as EPIA 2011. Although the analysis of the results was deeper in the referred article, it included only the experiments concerning the flat hierarchy experiment.

5.2 Future Work

Despite the interesting results, there are also many ideas for future improvements. For instance, the performance of our classification models is restricted by the availability of labeled training data, making it interesting to experiment with semi-supervised models, capable of leveraging on small sets of hand-labeled data together with large amounts of unlabeled data.

It will also be interesting to test other classifiers existent in the Weka Toolkit, in order to perceive if there are any other classifiers more suitable for the task of biographical sentence classification. Furthermore, it will be also interesting to observe if the performance of the voting classifiers increased with the increase of the number of classifiers used as voters. Moreover, the study of an optimum group of classifiers which could be used as voters could be important to improve the results, since it better to have few good voters than a lot of bad ones. Alternatively, the discovery of a method to assign properly different weights to different voters may achieve promising results.

Finally, it will be important to check if the conclusions of this work are independent of the type of the corpus origin or even its source language.

Bibliography

ABERDEEN, J., BURGER, J., DAY, D., HIRSCHMAN, L., ROBINSON, P. & VILAIN, M. (1995). Mitre: description of the alembic system used for MUC-6. In *Proceedings of the 6th Conference on Message Understanding*.

ABNEY, S. (1996a). Part-of-speech tagging and partial parsing. In *Corpus-Based Methods in Language and Speech*.

ABNEY, S. (1996b). Partial parsing via finite-state cascades. *Natural Language Engineering*, **2**.

APPELT, D. & ISRAEL, D. (1999). Introduction to information extraction technology. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*.

ARASU, A. & GARCIA-MOLINA, H. (2003). Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*.

BARISH, G., CHEN, Y.S., DIPASQUO, D., KNOBLOCK, C., MINTON, S., MUSLEA, I. & SHAHABI, C. (2000). Theaterloc: Using information integration technology to rapidly build virtual applications. In *Proceedings of the 16th International Conference on Data Engineering*.

BAUMGARTNER, R., FLESCA, S. & GOTTLOB, G. (2001). Visual web information extraction with lixto. In *Proceedings of the 27th International Conference on Very Large Data Bases*.

BLUNSOM, P., KOCIK, K. & CURRAN, J. (2006). Question classification with log-linear models. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

BOUCKAERT, R. & FRANK, E. (2004). Evaluating the replicability of significance tests for comparing learning algorithms. In *In Advances in Knowledge Discovery and Data Mining*.

BRANDOW, R., MITZE, K. & RAU, L. (1995). Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, **31**.

CHANG, C.H., CHUN-NAN, H. & LUI, S.C. (2003). Automatic information extraction from semi-structured web pages by pattern discovery. *Decision Support Systems*, **35**.

COHEN, W., HURST, M. & JENSEN, L. (2002). A flexible learning system for wrapping tables and lists in HTML documents. In *Proceedings of the 11th International Conference on World Wide Web*.

CONITZER, V. (2006). *Computational Aspects of Preference Aggregation*. Ph.D. thesis, Carnegie Mellon University.

CONWAY, M. (2007). *Approaches to Automatic Biographical Sentence Classification: An Empirical Study*. Ph.D. thesis, University of Sheffield.

COWIE, J. & LEHNERT, W. (1996). Information extraction. *Communications of the ACM*, **39**.

CUNNINGHAM, H. (2005). Information extraction, automatic. *Encyclopedia of Language and Linguistics*, **5**.

EDMUNDSON, H. (1969). New methods in automatic extracting. *Journal of the ACM*, **16**.

FERGUSON, N. (1992). *Variation Across Speech and Writing*. Cambridge University Press.

FERGUSON, N. (2000). *The Penguin Book of Journalism: Secrets of the Press*. London: Penguin.

FURNKRANZ, J. (1998). A Study Using N-Gram Features for Text Categorization. Technical Report OEFAI-TR-9830, Austrian Research Institute for Artificial Intelligence.

GONÇALO SIMÕES, L.C., HELENA GALHARDAS (2009). Information extraction tasks: A survey. In *INFORUM 2009 - Simpósio de Informática*.

GRISHMAN, R. (1997). Information extraction: Techniques and challenges. In *International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*.

GRISHMAN, R. & SUNDHEIM, B. (1996). Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics*.

HERMJAKOB, U., HOVY, E. & LIN, C.Y. (2002). Automated question answering in webclopedia: A demonstration. In *Proceedings of the 2nd International Conference on Human Language Technology Research*.

HOLMES, D. & FORSYTH, R. (1995). The federalist revisited: New directions in authorship attribution. *Literary and Linguistic Computing*, **10**.

HURST, M. (2000). *The Interpretation of Tables in Texts*. Ph.D. thesis, University of Edinburgh.

JACOBS, S. & RAU, L. (1990). Scisor: Extracting information from on-line news. *Communications of the ACM*, **33**.

KRUPKA, G. (1995). Sra: description of the sra system as used for MUC-6. In *Proceedings of the 6th Conference on Message Understanding*.

KUPIEC, J., PEDERSEN, J. & CHEN, F. (1995). A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

KWOK, C., ETZIONI, O. & WELD, D. (2001). Scaling question answering to the web. *ACM Transactions on Information Systems*, **19**.

LAFFERTY, J., MCCALLUM, A. & PEREIRA, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*.

LEWIS, D. (1992). *Representation and Learning in Information Retrieval*. Ph.D. thesis.

LI, X. & ROTH, D. (2002). Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*.

LIU, Y., BAI, K., MITRA, P. & GILES, C. (2007). Tableseer: Automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS joint Conference on Digital libraries*.

LUHN, P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, **2**.

MANI, I. (2001). Summarization evaluation: An overview.

MANI, I. & MACMILLAN, T.R. (1996). *Identifying Unknown Proper Names in Newswire Text*. MIT Press.

MARCU, D. (1999). The automatic construction of large-scale corpora for summarization research. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

MCCALLUM, A. (2005). Information extraction: Distilling structured data from unstructured text. *Queue*, **3**.

MCCALLUM, A. & NIGAM, K. (1998). A comparison of event models for naive bayes text classification. In *Proceeding of the AAAI/ICML-98 Workshop on Learning for Text Categorization*.

MENDES, A., COHEUR, L., MAMEDE, N., RIBEIRO, R., BATISTA, F. & MATOS, D. (2008). *QA@L2F, First Steps at QA@CLEF*, 356–363. Springer-Verlag, Berlin, Heidelberg.

MILLER, S., CRYSTAL, M., FOX, H., RAMSHAW, L., SCHWARTZ, R., STONE, R. & WEISCHEDEL, R. (1998). Algorithms that learn to extract information: Bbn: Tipster phase iii. In *Proceedings of a Workshop on Held at Baltimore*.

MOGUERZA, J.M. & MUÑOZ, A. (2006). Support vector machines with applications. *Statistical Science*, **21**.

MOLDOVAN, D., PAŞCA, M., HARABAGIU, S. & SURDEANU, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems*, **21**.

PAICE, C. & JONES, P. (1993). The identification of important concepts in highly structured technical papers. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

PAN, Y., TANG, Y., LIN, L. & LUO, Y. (2008). Question classification with semantic tree kernel. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

PANG, B. & LEE, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, **2**.

PAPE, S. & FEATHERSTONE, S. (2005). *Newspaper Journalism: A Practical Introduction*. Sage, London.

PINTO, D., MCCALLUM, A., WEI, X. & CROFT, W. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

RATH, G., RESNICK, A. & SAVAGE, T. (1961). The Formation of Abstracts by the Selection of Sentences. Part I. Sentence Selection by Men and Machines. **12**.

REIMER, U. & HAHN, U. (1988). Text condensation as knowledge base abstraction. In *Proceedings of the 4th Conference on Artificial Intelligence Applications*.

RIFKIN, R. & KLAUTAU, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research*, **5**.

SANTINI, M. (2004). A shallow approach to syntactic feature extraction for genre classification. In *In 7th Annual Computer Linguistics UK Research Colloquium*.

SARAWAGI, S. (2008). Information extraction. *Foundations and Trends in Databases*, **1**.

SCHIFFMAN, B., MANI, I. & CONCEPCION, K. (2001). Producing biographical summaries: Combining linguistic knowledge with corpus statistics. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*.

SEBASTIANI, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**.

SILVA, J. (2009). *QA+ML@Wikipedia&google*. Master's thesis, Instituto Superior Técnico.

SODERLAND, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, **34**.

STAMATATOS, E., KOKKINAKIS, G. & FAKOTAKIS, N. (2000). Automatic text categorization in terms of genre and author. *Computational Linguistics*, **26**.

TAIT, I. (1985). Generating summaries using a script-based language analyser. In *Selected and Updated Papers From the Proceedings of the 1982 European Conference on Progress in Artificial Intelligence*.

TSUR, O., RIJKE, M. & SIMA'AN, K. (2004). Biographer: Biography questions as a restricted domain question answering task. In *Proceedings on Association for Computational Linguistics Workshop on Question Answering in Restricted Domains*.

XIAO, Z. & MCENERY, A. (2005). Two approaches to genre analysis: Three genres in modern american english. *Journal of English Linguistics*, **33**.

ZHOU, L., TICREA, M. & HOVY, E. (2005). Multi-document biography summarization. *Computing Research Repository*.