TÉCNICO LISBOA

Extracting Relationships and Network Structures from Text

David José Martins Forte

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Bruno Emanuel da Graca Martins

Examination Committee

Chairperson: Prof. José Manuel Nunes Salvador Tribolet
Supervisor: Prof. Bruno Emanuel da Graca Martins
Member of the Committee: Prof. Maria Luísa Torres Ribeiro Marques da Silva Coheur

November 2014

Acknowledgments

Quero dirigir as minhas primeiras palavras a algumas pessoas em particular, pela ajuda imprescindível ao longo do tempo em que trabalhei na minha tese de mestrado. Gostaria de comecar por agradecer ao meu orientador Professor Bruno Martins, pela sua ajuda, por todo o conhecimento demonstrado, rigor científico, capacidade de trabalho, entusiasmo, empenho e disponibilidade que sempre revelou, e sem os quais esta fase final do curso nao teria sido possível. Um agradecimento também aos meus colegas do grupo DMIR do INESC-ID, pela sua ajuda nas fases iniciais do meu trabalho. Gostaria ainda de agradecer o suporte financeiro da Fundacao para a Ciencia e Tecnologia (FCT), atraves de bolsas nos projectos REACTION (UTA-Est/MAI/0006/2009) e KD-LBSN Knowledge Discovery from Location Based Social Networks (EXPL/EEI-ESS/0427/2013). Por fim, deixo tambem um especial agradecimento a todos os que me acompanharam ao longo deste percurso, tanto aos que estiveram directamente envolvidos durante o processo, como aos que sem estarem directamente envolvidos no meu percurso academico, me apoiaram incondicionalmente.

Resumo

O facto de informação disponível na internet aumentar exponencialmente de dia parta dia, torna cada vez mais interessante a investigação de temas como extração de relações entre entidades em documentos de texto, nomeadamente entre pessoas e entre locais. Este trabalho de investigação está dividido em dois objectivos. O primeiro está relacionado com o estudo de técnicas que permitam extração de relações em texto escritos em Portugûes, apontando ao desenvolvimento de um sistema capaz de extrair relações de polaridade entre pares de pessoas. O segundo objectivo, refere-se á utilização desse mesmo sistema, com algumas adaptações, para efectuar extração de relações entre pares de locais, mencionados em livros de ficção escritos em Inglês, onde um local é ”parte de” outro (por exemplo, Lisboa ”parte de” Portugal). Este trabalho, refere também uma abordagem para reconhecimento de entidades em textos escritos em Português, que será utilizada para identificar as pessoas nos textos utilizados no primeiro objectivo.

Esta dissertação formaliza assim abordagens para reconhecimento de entidades mencionadas e extração de relações, apresentando um eficiente sistema para reconhecimento de entidades em documentos de texto em Português. É ainda apresentada uma descrição do sistema de extração de relações que foi desenvolvido. Este sistema, é a extensão de um algoritmo proposto anteriormente, o Snowball. É também efectuada uma validação experimental às duas versões do sistema de extração de relações, utilizando o jornal Público para a extração de relações de suporte e oposição e livros de geografia e ficção para a extração de relações entre locais.

Palavras-chave: Extração de Relações, Contrução de Redes a partir de Texto, Reconhecimento e Desambiguação de Entidades

Abstract

Given the increasing availability of textual contents on the Web, it is nowadays be interesting to extract analyze the relations between persons and between locations that are mentioned in textual documents.

My research work had two main objectives. The first objective was to study relation extraction techniques for the Portuguese language, aiming at the development of a system for extracting polarity relations between pairs of persons from collections of textual documents, through techniques from the areas of Information Extraction and Natural Language Processing. The second objective was to use the same relation extraction techniques, aiming at the extraction of relations between pairs of locations mentioned in fiction works written in English language, where a location is part of another. In order to achieve the two main objectives, I first needed to address the task of recognizing the person and locations entities in the documents.

This dissertation thus formalizes my approaches to the named entity recognition and relation extraction problems, describing the most relevant related work, presenting an efficient and robust Named Entity Recognition system for the Portuguese language, and presenting a description of a relation extraction system that was developed as an extension of the previously proposed Snowball algorithm, capable of extracting support and opposition relations between two persons, and part-of relations between two locations. It also presents an extensive experimental validation that has been carried out with different configurations of the system, using data from a Portuguese newspaper, Público, using a geography book of United Kingdom, and fiction books.

Keywords: Relation Extraction, Building Networks from Text, Entity Recognition and Disambiguation

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The analysis of quantitative information derived from document collections, such as daily newswire texts, e-book libraries or social media contents, holds an enormous potential to solve long standing problems in a variety of disciplines, through massive data analysis [Zhu, 2010]. For instance political scientists, sociologists or historians can all stand to benefit significantly from massive analysis of newswire documents, as these fields are primarily concerned with studying events involving entities that are widely covered in the media. Recently, we have been indeed witnessing an increasing interest on the usage of techniques from the areas of text mining, information extraction and natural language processing (NLP), in applications related to the social sciences, the digital humanities, and to media analytics in general [Ryu et al., 2012]. Information extraction, in particular, can be divided into several sub-tasks, and some of the most relevant sub-tasks are named entity recognition, named entity disambiguation, and relation extraction. These three tasks can be used together to generate new knowledge about entities that are referenced within texts.

- **Named Entity Recognition** refers to the identification of named entities as referenced over textual documents, and to their classification into one of several entity types, such as person, organization or location. For instance, in the sentence *Passos Coelho is the prime minister of Portugal.*, the string *Passos Coelho* should be recognized as a named entity, and labeled with the type person. The string *Portugal* that should also be recognised and labeled with the type location.

- **Named Entity Disambiguation** refers to the task of assigning an identifier to an entity mention, previously recognized by a named entity recognition system, that represents that entity in the real world. For example, the named entities *Passos Coelho* and *Pedro Passos Coelho* represent the same real world entity, and so they should have the same identifier assigned to them, even if occurring in different documents.

- **Relation Extraction** concerns with the identification of relations between two or more entities previously recognized in a text. For example, in the sentence *Passos Coelho is a member of the Portuguese Parliament.*, a relation between the named entities *Passos Coelho* and *Portuguese Parliament* should be identified, and perhaps later categorized as a *member* of relation, between

a specific person and an organization.

My MSc thesis concerns with named entity recognition and relation extraction from text, combining ideas from the areas of text mining and network analysis. A particular experiment performed in the context of this work involved the extraction of all positive and negative evaluative opinions, as expressed by particular persons in a given collection of textual documents (e.g., news articles) about other persons, thus supporting the generation of graphs encoding positive and negative interactions between persons. A second experiment was instead concerned with relation extraction between locations, specifically the extraction of part-of relations from the textual contents of books.

1.1 Hypothesis and Methodology

My MSc research effectively tried to prove, through controlled experiments with prototype systems, three main hypothesis related to the development of information extraction systems, namely:

1. The relation extraction task can be addressed successfully in the case of documents written in Portuguese, and when considering the specific task of extracting support and opposition relations between persons, using bootstrapping techniques for performing the extraction of relations between individuals, through an adapted version of the previously proposed Snowball approach.

2. Bootstrapping techniques for relation extraction can be effective in the extraction of part-of relations between pairs of location entities, in the case of documents written in English.

In order to evaluate the first aforementioned hypothesis, I created two ground-truth datasets based on the Portuguese parliament (e.g. politicians in the same or in opposite parties), in order to support the realization of experiments and the measurement of results, using common metrics from the area of information extraction (e.g., precision, recall, F_1 measure, and accuracy). The part-of relations were evaluated by using the same metrics and a dataset containing all the part-of relations regarding the *United Kingdom*. I created this dataset based on the GeoPlanet[1] service, that provides an open, permanent, and intelligent infrastructure for geo-referenced data on the Internet.

From the creation of the relation extraction systems, and also from the extensive set of experiments that were performed, a series of important contributions were achieved, which are described in the following section.

1.2 Original Contributions

The research made in the context of my MSc thesis led to the following main contributions:

1. I created and evaluated Named Entity Recognition (NER) models for the Portuguese language, using the StanforNER[2] software framework, in order to identify the named entities present in a

[1] `https://developer.yahoo.com/geo/geoplanet/`
[2] `http://nlp.stanford.edu/software/CRF-NER.shtml/`

document written in the Portuguese language, prior to their disambiguation or the extraction of relations involving these entities. The dataset used in the training of this model was the CINTIL[3] corpus of modern Portuguese. The model was tested using a 5-fold cross-validation technique, where a maximum of 68.06% F_1 was achieved, when evaluating the models in terms of detecting the correct entity spans.

2. I developed an heuristic method for entity disambiguation of person names, in order to improve the efficiency of the recognition of persons which can be mentioned with different names. In this procedure, I used heuristics such as dividing the persons by gender, using male and female names, removing labels present in the entity name, and only disambiguating names with more than one appearance in the texts. After an initial filtering, names are compared with all the other names associated to the same gender, using a simple similarity procedure based on the Jaro-Winkler TF-IDF similarity measure.

3. I implemented a relation extraction system that relies on the algorithm named *Snowball*, introducing several adaptations. This system uses a documents collection and a set of seeds to extract relations between person entities, and relations between location entities. The differences from the original algorithm relate to an improvement of the confidence formulas, the use of another clustering methodology, and the application of the structural balance theory to predict and evaluate existing relations.

4. I built two different ground-truth datasets for evaluating the extraction of support and opposition relations, based on an automated procedure that leverages a list with all the Portuguese politicians which seat in the parliament, and the respective parties to which they belong, assuming that:

    - Political entities that belongs to parties with the same orientation (i.e., left or right) have a support relation between them, and they have opposition relations towards the persons with the other orientation. The system was tested with this dataset, using the evaluations metrics used in the area, achieving a maximum of 69.26% F_1.
    - Political entities that belongs to the same party have a support relation between them, and a opposition relation towards the persons on the other parties. When testing with this dataset, the results showed a F_1 maximum of 77.83%.

5. I created a dataset containing part of relations between locations, by using the GeoPlanet[4] service. The results in this case returned a F_1 of 19.2%.

1.3 Outline for the Dissertation

The rest of this document is organized as follows: Chapter 2 describes important concepts involved in understanding the proposed work, and presents previous related work addressing the relation extraction

[3] http://cintil.ul.pt/
[4] https://developer.yahoo.com/geo/geoplanet/

task, namely works that used supervised methods, others that rely on unsupervised approaches, and works that have specifically focused on building signed networks from text. Chapter 3 describes the importance of named entity recognition in information extraction, explaining the techniques that I used to address this particular task. This chapter also shows the experimental results obtained with the proposed method, as well as the conclusions taken from this task. Chapter 4 presents the relation extraction system that I developed to extract support and opposition relations, giving an overview of its architecture and describing in detail the named entity disambiguation stage, and how the system extracts support and opposition relations in order to build signed networks. Finally, the results are presented, describing the experiments that measured the system's accuracy over Portuguese texts, together with the conclusions taken from the development of the relation extraction system. Chapter 5 presents a relation extraction system designed to extract part of relations between location entities. The system consists on a adaptation of the extraction system explained in the previous chapter. This chapter also presents the experimental validation, describing evaluation experiments over English texts, and the conclusions drawn from the experiment. Finally, Chapter 6 presents the main conclusions from this research, and discusses possible directions for future work.

Chapter 2

Concepts and Related Work

This chapter presents the main concepts involved in understanding the work that I have made in the course of developing my MSc thesis. It also presents a brief overview on research works related with the tasks of relation extraction from text, opinion mining, the extraction of social networks from text, and sentiment slot filling.

2.1 Fundamental Concepts

This section presents fundamental concepts, explaining how textual information can be treated for computational processing, describing the mechanisms that are typically used to represent textual documents, describing how one can identify the entities mentioned in texts, as well as how one can extract and analyze the interactions and relationships between them, and presenting the algorithms that are traditionally used for text classification.

2.1.1 Representing Textual Information

The choice of an adequate representation for textual information is the starting point for any text analysis task. Each document, in a given dataset, can be seen as being made of simpler units, and in the context of tasks such as document classification or information extraction, documents are typically represented through sets or vectors of such smaller units, like words or bi-grams of words.

When we have a document collection to be used on some text analysis task, the first thing to do is splitting the individual documents into sentences. This task is known as sentence splitting, and it typically uses simple heuristics for breaking the text (e.g., the sentences can be separated by punctuation marks or line breaks). After sentence splitting, we also need to identify the individual words. The act of breaking up a sequence of characters into meaningful simpler units, such as words, is known as tokenization. The resulting tokens become the input for subsequent text processing tasks.

Tokenization also relies oftenly on simple heuristics, in which the most basic of them involves having words being separated by whites spaces, punctuation marks, or line breaks. The generated tokens can be made up entirely of alphabetic characters, alphanumeric characters, or numeric characters only.

There are some languages that use inter-word spaces or hyphens, and one needs to be careful with that. For instance, cases such as contractions, hyphenated words, and larger constructs such as URIs or URLs, should ideally be properly accounted for.

For this work, I used a sentence splitter and a tokenizer from the OpenNLP[1] toolset. The sentence splitter that I have used relies on a maximum entropy model to evaluate the characters like `.`, `!`, and `?` in a given sequence of characters, in order to determine if they signify the end of a sentence. For tokenization, OpenNLP in fact offers multiple implementations, namely:

1. **A Simple Whitespace Tokenizer:** Non white space sequences of characters are identified as tokens.

2. **A Simple Character-Class Tokenizer:** Sequences of the same character class are used to form the tokens.

3. **A Learnable Tokenizer:** A maximum entropy tokenizer, which detects token boundaries based on a probabilistic model.

OpenNLP has some specific tools which can be used to train sentence splitting and tokenization models. For training models specific to Portuguese texts, I specifically used training data from the CINTIL Corpus for Modern Portuguese[2] [Barreto et al., 2006]. For the English case, I used the learned models that are already distributed with OpenNLP.

After texts have been segmented into individual tokens, we can use the tokens to build a representation for the documents, for instance through the vector space model. The vector space model is an algebraic model for representing text documents as vectors of identifiers (i.e., tokens). It is widely used in information filtering and information retrieval, allowing us to compute a continuous degree of similarity between documents.

Each dimension of the vector representation of a document corresponds to a separate token occurring in the collection. If a token occurs in the document, its value in the vector is non-zero. The dimensionality of the vector is thus the number of tokens in the vocabulary.

Salton et al. [1975] proposed a vector space model in which the term-specific weights in the document vectors are products of local and global parameters. The model is known as the Term Frequency times Inverse Document Frequency model (TF-IDF). As the name indicates, TF-IDF combines the individual frequency for each element i in a specific document j (i.e., a TF component corresponding to the term frequency), with the inverse frequency of element i in the entire collection of documents (i.e., an IDF component corresponding to the inverse document frequency). In the case of the term frequency $\mathrm{tf}_{t,d}$, the simplest choice is to use the frequency of a term in a document, i.e. the number of times that term t occurs in document d. In the complete TF-IDF scheme, the vector representation of a document d is given by $\mathbf{v_d} =< w_{1,d}, w_{2,d}, \ldots, w_{N,d} >$, where N is the number or features (i.e., the number of different tokens in the collection), and where the weights $w_{t,d}$ correspond to:

[1] `http://opennlp.apache.org/`
[2] `http://cintil.ul.pt/pt/cintilfeatures.html`

$$w_{t,d} = \mathrm{tf}_{t,d} \cdot \log\left(\frac{|D|}{|\{d' \in D \,|\, t \in d'\}|}\right) \tag{2.1}$$

In the formula, $\mathrm{tf}_{t,d}$ is term frequency of term t in document d, $|D|$ is the total number of documents in the document set, and $|\{d' \in D \,|\, t \in d'\}|$ is the number of documents containing the term t.

Considering vectorial representations for the documents, one can compute a similarity coefficient between document pairs, which reflects the degree of similarity in the corresponding terms and term weights. Such a similarity measure might be the inner product of the two vectors, or alternatively an inverse function of the angle between the corresponding vector pairs (e.g., the cossine of the angle). When the term assignment for two vectors is identical, the angle will be zero, producing a maximum value of the similarity measure.

2.1.2 Natural Language Processing and Linguistic Annotation

Natural Language Processing (NLP) explores how computers can be used to understand and manipulate natural language texts to do useful things. NLP researchers gather knowledge on how human beings understand and use language, so that automated tools and techniques can be developed. Applications of NLP include machine translation, summarization, and information extraction.

Typically, NLP is based on a pipeline of operations that involves tokenization, sentence splitting, part-of-speech tagging, parsing, named entity recognition, and relation extraction. The idea is that the output of one task is the input of the next one. This pipeline can be composed by all of the tasks mentioned above, or just by some of them. The following subsections detail each of the most important steps, except for tokenization and sentence splitting, which were already covered in the previous section.

Parts-of-Speech Tagging

Parts-of-Speech (POS) tagging is the process of marking up the words in a text as corresponding to particular morphological classes, based on their definition and context (i.e., based on the words themselves and on the relationship with adjacent and related words in a given sentence). Through POS tagging, one can know the gramatical class (e.g. noun, verb, adjective, adverb, etc.) of each word.

Parts-of-speech tagging is a relatively challenging task, for which it is not enough to just have a list of words and their most common parts of speech, because some words are complex and can correspond to more than one part-of-speech at different times (i.e., a large percentage of word-forms are ambiguous). For example, in the Portuguese case, the word *cedo* (i.e., early) is usually thought of as just an adverb, although it can also be a specific form of the verb *ceder* (i.e., to give in).

POS tagging has been addressed with relative success for a variety of languages, and the set of POS tags that is used varies greatly with each language. Tags usually are designed to include overt morphological distinctions, although this may lead to inconsistencies. Recent work in the area has advocated for the use of general coarse-grained POS tagsets in NLP applications, which can be shared across multiple languages [Petrov et al., 2012].

The OpenNLP toolset can be used to perform POS tagging, with basis on a maximum entropy probabilistic model. For representing each word, OpenNLP generates a binary characteristics vector were each position represents a possible token characteristic (e.g., the actual word in a given position of the sentence, a lowercased version of the word, an indication on if the token is alpha-numeric, a value indicating if the initial character is a capital letter, a value indicating if we are in the beginning of a sentence, etc.). The tag assigned to each word is calculated with basis on a maximum entropy model that uses the characteristics vector. The maximum entropy model will be explained in more detail in Section 2.1.3.

In order to find the best sequence of tags for a sequence of words, we need to consider the probability values of the previous word categories because, sometimes, the tag of the previous words can influence the tag of the word that we want classify. For this we can use the Viterbi algorithm, that is a dynamic programming algorithm for finding the most likely sequence of classes in a sequence of observed events. Using the Viterbi algorithm, we test all the possible sequences of tags that a sequence of words can have, and then we choose the sequence of tags with higher probability of occurrence.

Named Entity Recognition

A named entity is a word, or compound of words, that clearly identifies a real world entity, including persons, organizations and geographic locations. Identifying references to these entities, in textual documents, is one of the most important sub-tasks of information extraction, and this is typically called Named Entity Recognition and Classification (NERC).

NER systems have been created by using linguistic grammar-based techniques, as well as with statistical models. Statistical systems typically require a large amount of manually annotated training data.

We can model NER as a task of giving tags to words, much like POS tagging. Most statistical NER systems rely on the so-called IOB encoding, where the classifiers are trained to recognize the beginning (B), the inside (I), and the outside (O) of an entity name. Consider the following example:

David/**B-PER** Forte/**I-PER** studies/**O** in/**O** IST/**B-ORG** ./**O**

In the example, we have two different types of entities, namely a person and an organization. The first entity is composed by more than one word, and thus the first word is assigned to a B(eginning) tag and the other words are assigned to a I(nside) tag. When the entity is composed by one word, it is always assigned a B(eginning) tag, such as in the second entity. The other words in the example have a tag O(ther), because they are not entities.

NER systems can be developed through maximum entropy models, which will be explained in more detail in Section 2.1.3. The Viterbi algorithm can again be used to find the best sequence of tags for a sequence of words. NER models can use features such as the actual word, the previous word, indicators for if the word is in lowercase or if the first letter is in uppercase, lists of known entities (i.e. if the word is in a list of names, then the probability of this word being a name is very high), etc.

In my work, for training NER models, I used the training data from the CINTIL Corpus of Modern Portuguese, but with the Stanford NER toolkit instead of OpenNLP. This decision is related with the

computational performance of the two different implementations. In the recognition of the named entities in one sentence, a model trained with OpenNLP takes much more time than a Stanford NER model, and for a large amount of sentences, the difference in processing time is huge. For named entity recognition in English texts, used the model that is distributed with Stanford NER.

Relation Extraction

The identification of semantic relationships, as expressed between named entities in text, is another important step for extracting knowledge from document collections. A relationship extraction task is oftenly formulated as the detection and classification of semantic relationship mentions within a sentence. We therefore have that similarity to the previous tasks, relation extraction can also be represented as a classification task, with the difference that instead of assigning a class to a word, we assign a class to a pair of entities. Consider, for example, the following sentence:

> **Sintra** is a village in the region of **Lisbon**, and **David** lives there.

When considering the previous example, we can start by identifying the pairs of entities in the sentence: Sintra-Lisbon, Sintra-David, and Lisbon-David. To each pair of entities, we will associate a binary feature vector where each position of the vector represents a different feature (e.g. if the fist entity is a person, if between the two entities there is a specific word, etc.). With this feature vector, we want to find the specific class of the relation, and for this we can for instance use a maximum entropy model that will calculate which is the category with the higher probability. The category can be *located-in*, *influenced-by*, *successor-of*, etc. In this example, we have a semantic relationship of the type *located-in* expressed between two place names (i.e., *Sintra* and *Lisbon*), and another relation of the same class between *Sintra* and *David*.

2.1.3 Text Classification

Many problems in Natural Language Processing (NLP) can be formulated as statistical classification problems. Such classifications can be made through a maximum entropy model, a technique that offers a clean way to combine diverse pieces of contextual evidence in order to estimate the probability of a certain class occurring with a certain context.

Maximum entropy models use specific characteristics of the instance that we want to classify. These characteristics are called features. More formally, we want to estimate $\mathrm{P}(c|x)$, i.e the probability of a class c given a context x.

The following formula is calculated for all the categories, and the respective instance is assigned to the category with the highest probability value.

$$\mathrm{P}(c|x, \boldsymbol{\lambda}) = \frac{\exp \sum_i \lambda_i \mathrm{f}_i(c, x)}{\sum_{c' \in C} \exp \sum_i \lambda_i \mathrm{f}_i(c', x)} \tag{2.2}$$

In Formula 2.2, $\mathrm{P}(c|x, \boldsymbol{\lambda})$ represents the probability that a category c can have, given an instance x and a set of feature weights $\boldsymbol{\lambda}$. We calculate the exponential function for the sum of each feature applied to the instance x and category c, multiplied by the respective weight of the feature, dividing this by the same exponential function applied on all the categories. This formula is applied to all the categories and the category with higher probability is the category assigned to x. The weight of the features is calculated during model training by optimizing a convex objective function through and algorithm such as L-BFGS [Malouf, 2002, Liu and Nocedal, 1989].

Another approach for text classification is the k-nearest neighbours algorithm (kNN), which predicts class memberships based on the k closest neighbours of the instance. The neighbours are taken from a set of instances for which the correct classification is known. In this case, the training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a defined constant, and an unlabeled vector is classified by assigning the label which is most frequent among the k training samples nearest to that query point.

A particular problem with nearest neighbour classification approaches concerns with efficiently estimating the tok-k most similar training instances. Min-hash is a technique for quickly estimating how similar two sets of elements are, according to the Jaccard similarity coefficient [Broder, 1997]. Min-hash can thus be naturally used to implement nearest neighbour classifiers.

Given a vocabulary Ω and two sets of elements $S_1, S_2 \subseteq \Omega$, we have that the Jaccard similarity coefficient between the two sets is the size of the intersection of S_1 and S_2, divided by the size of the union of S_1 and S_2, as shown in Formula 2.3.

$$\mathrm{J}(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}. \tag{2.3}$$

The Jaccard similarity coefficient returns a value between 0 and 1, where 0 means that the two sets are disjoint and 1 means they are equal (i.e., two sets are more similar when their Jaccard index is closer to 1, and more dissimilar when their Jaccard index is closer to 0). Computing the Jaccard similarity coefficient is relatively simple, but it is also computationally heavy, giving that we need to compute the intersection between large sets. A solution for this is the use of the min-hash technique. Noticing that the Jaccard coefficient corresponds to the number of elements that occur in S_1 and S_2 simultaneously, divided by the the number of elements that occur at least once in one of the sets, we have that a probabilistic interpretation for the similarity coefficient corresponds to:

$$\Pr[\mathrm{h}_{min}(S_1) = \mathrm{h}_{min}(S_2)] = \mathrm{J}(S_1, S_2). \tag{2.4}$$

In the min-hash mechanism, if we think about $\mathrm{h}(S)$ like a hash function that maps the members of S_1 and S_2 to distinct integers, and if for any set S we define $\mathrm{h}_{min}(S)$ as the member of S with the minimum value of $\mathrm{h}(S)$, then $\mathrm{h}_{min}(S_1) = \mathrm{h}_{min}(S_2)$ happens exactly when the minimum hash value of the union $S_1 \cup S_2$ lies in the intersection $S_1 \cap S_2$. In other words, if r is a random variable that is one when $\mathrm{h}_{min}(S_1) = \mathrm{h}_{min}(S_2)$ and zero otherwise, then r is an estimator for the Jaccard coefficient.

The simplest version of the min-hash scheme uses k different hash functions, where k is a fixed integer parameter, and it represents each set S by the k values of $\mathrm{h}_{min}(S)$. To estimate $\mathrm{J}(S_1, S_2)$, if we let y be the number of hash functions for which $\mathrm{h}_{min}(S_1) = \mathrm{h}_{min}(S_2)$, then we can use $\frac{y}{k}$ as the estimate for the Jaccard similarity coefficient.

An efficient way to search efficiently for the kNN can be implemented through a hashing technique that is sensitive to the similarity between the instances, called Locality Sensitive Hashing (LSH). This technique uses the min-hash signatures to compress the representation of the elements into small signatures, preserving at the same time the expected similarity for any pair of instances. The algorithm uses L different dispersion tables, each one corresponding to a n-tuple of hash signatures, called band. When classifying a specific instance, we calculate the min-hash signature of this instance, and then consider any instance associated to a same bucket of the data structure, for any min-hash band, as a candidate instance to belong to the set of kNN most similar instances. The candidate pairs are checked using the complete hash signatures to approximate the Jaccard similarity coefficient. With this, we avoid the similarity comparison between all the elements in the data base.

2.2 Relation Extraction From Text

As explained in Section 2.1.2, extracting semantic relationships between nominal expressions (e.g., between named entities) is an important step in natural language understanding. Several authors have proposed machine learning techniques to address this problem, for instance formulating it as a binary supervised classification task defined on candidates to nominal expression pairs, where the candidates are classified as *related* or *not related*. This approach also be easily extended to the case of n different types of semantic relations between entities.

Kambhatla [2004] employed maximum entropy models for extracting relations, combining diverse lexical, syntactic and semantic features. The authors used the list of relation types and subtypes of the ACE 2003 evaluation [Doddington et al., 2004]. Entities were of five types: person, organization, location, facilitie, or geo-political entity. They also used ACE training data to train the relation extraction model, containing 5 different relation types and 24 sub-types (e.g., type SOCIAL with sub-types *associate*, *parent*, etc., or type AT with the sub-types *based-in*, and *located*).

The authors explicitly modelled the argument order of mentions. Thus, when comparing mentions p_1 and p_2, they distinguish between the case where p_1-*citizen-Of*-p_2 or p_2-*citizen-Of*-p_1. The extraction was formulated as a classification problem with 49 classes, two for each relation subtype and a NONE class for the case where the two mentions are not related. For each pair of mentions, they compute several features, such as token identity, entity type, mention-level features (i.e., one of NAME, NOMINAL, PRONOUN of both the mentions), the number of words separating the two mentions, words and parts-of-speech tags of these words, and the path of non-terminals connecting the two mentions in a parse tree for the corresponding sentence. Their results indicate that using a variety of information sources can result in an improved recall and overall F_1 measure. This approach can also easily be adapted to use more features (e.g., from sources like WordNet, gazatteers, output of other semantic taggers etc.).

Besides methods based on supervised learning for semantic relation extraction in a specific domain, there are some research works that focused on relation extraction in an open context, such as TextRunner[3], ReVerb[4], OLLIE[5] or SOFIE[6]. Some of the techniques that are in the base of these systems are described on a article about Open Domain Information Extraction (Open IE) by Etzioni et al. [2008]. Open IE is an extraction paradigm that tackles an unbounded number of relation types, avoids the need for domain-specific training data, and scales linearly to handle Web-scale corpora. For example, an Open IE system might operate in two phases. First, it would learn a general model of how relations are expressed in a particular language. Second, it could utilize this model as the basis of a relation extractor whose sole input is a corpus and whose output is a set of extracted tuples that are instances of a potentially unbounded set of relations. Such an Open IE system would learn a general model of how relations are expressed (in a particular language), based on features such as parts-of-speech tags (for example, the identification of a verb in the surrounding context can offer important clue to the relation type) and domain-independent regular expressions (for example, the presence of capitalization and punctuation). However, domain-independent approaches generally produce worse quality results, and do not normalize the relations extracted.

To avoid the difficulty associated to the manual annotation of training data, some authors explored alternative paradigms for relation extraction from texts, based on distant supervision, bootstrapping or other methods. For instance, Mintz et al. [2009], Krause et al. [2012], or Riedel et al. [2010] used Freebase[7], a structured database of semantic information covering thousands of relations between entities, as a way to construct the training examples. For each pair of entities referenced in a Freebase relation, the authors searched for sentences in a corpus that contain the entities, using these sentences as training examples for a relation extractor based on a traditional classifier. With this, the authors combine the advantages of supervised methods to relation extraction, with the advantages of non-supervised open-domain methods.

For instance, in the case of the experiment made by Mintz et al. [2009], the authors used a maximum entropy classifier combining lexical attributes (e.g., sequences of words and correspondent POS tags) and syntactic attributes (e.g., dependences between the entities involved in the relation). The results showed that the distant supervision method, based on Freebase, allowed for the extraction of 10.000 copies of 102 different types of relations, with a precision of 67.6%.

García and Gamallo [2011] compared the impact of different types of linguistic characteristics (sets of lemmas and POS tags, lexical-syntactic patterns, and syntactic dependences) in the task of relation extraction, through machine learning and using a technique that relies on distant supervision, following the method of Mintz et al. [2009], and using a collection of training samples collected from Wikipedia texts. The results showed that characteristics based on lexical-syntactic patterns obtain a greater precision than characteristics based on sets of words or syntactic dependences, although the combination of different types of characteristics allows for a better commitment between precision and recall, improving

[3] http://www.cs.washington.edu/research/textrunner/
[4] http://reverb.cs.washington.edu/
[5] https://github.com/rbart/ollie
[6] http://www.mpi-inf.mpg.de/yago-naga/sofie/
[7] http://www.freebase.com/

the performance of models that just use one type of characteristics. The experiments also showed that models that use lexical-syntactic patterns based on the middle context (in other words, models that just use words that occur between the related entity pairs) have better performance that those using all the contexts, i.e., the information of words that occur before, between and after the mentioned entities.

In a subsequent study, Gamallo et al. [2012] addressed the task of relation extraction between entity pairs using texts from different versions of Wikipedia (i.e., Portuguese, Spanish, English and Galician) and using a multi-language syntactic analyzer based on rules. The method involves three steps, namely (1) perform a dependency analysis, where each sentence in the input text is processed by a tool named TreeTagger[8] in order to assign POS tags to the words, and is then analyzed by a syntactic parser named DepPattern[9], in order to find dependences, (2) find for each analyzed sentence the verbal clauses and their constituents, including their functions (e.g. subjects, direct objects, attributes and prepositional complements), and (3) apply rules, using patterns on the constituent clauses that contain just one subject and one direct object. Evaluation tests showed that the system achieves 68% precision, but is has a low recall because of the low coverage of the grammars it depends on. To test the system's speed, the authors ran it over 100,000 lines of the English Wikipedia, and the processing time was of 5 minutes and 19 seconds.

Specifically on what concerns relation extraction in Portuguese texts, some results were also obtained in the ReRelEM sub-task of HAREM [Freitas et al., 2008], which had the objective of evaluating the recognition and the classification of semantic relations between the entity pairs mentioned in a given document. The three different systems that participated in the ReRelEM task opted for the recognition of different types of relations.

The SeRELeP system, which was one of the participants, used heuristic rules on the results of a syntactic analyzer, named *palavras* [Bick, 2000], recognising relations of the types identity, occurrence and inclusion between the mentioned entities [Bruckschen et al., 2008]. For instance, in the sentence *Brigada Militar de Porto Alegre*, the system would recognise a semantic relation of the type occurrence through the rule: if there exists a mentioned location (e.g. *Porto Alegre*) and this entity is part of another mentioned entity of the type organization (e.g *Brigada Militar de Porto Alegre*), then the entity is marked as related with the entity of the type organization.

The participant system named REMBRANDT, obtained the best task results, having achieved a F_1 measure of 45.02%, using basic heuristics of relationships between entities based on it's units and categories [Cardoso, 2008].

Batista et al. [2013] proposed to check which semantic relationship is expressed between a given pair of entities referenced in a sentence, with basis on an approximation to the Jaccard similarity coefficient, obtained through min-hashing and using also a Locality-Sensitive Hashing (LSH) technique in order to quickly find the kNN most similar relationship instances. The authors used training examples corresponding to sentences that express semantic relationships between pairs of entities. These examples were automatically collected from Wikipedia articles that referenced resources known to be related

[8] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
[9] http://gramatica.usc.es/pln/tools/deppattern.html

in DBPedia (i.e., the relations in DBPedia were combined with the sentences presented in Wikipedia's articles for the entities involved). The examples are represented as sets of character quad-grams and other representative elements (e.g., prepositions, verbs, and relational lexical-syntactic patterns) mostly extracted through POS tagging. The min-hash signatures are extracted from these sets of character quad-grams, and are divided in LSH hash bands. To classify new entity pairs according to their relationships, they extracted the quad-grams of the sub-sequences that occur before, after, and between the involved entities, and the min-hash signature of these sub-sequences is generated. Finally, the example relationships with at least one identical LSH hash are considered as candidates, and their similarly with the relation instance that we want to classify is then estimated using the min-hash signatures. The most similar examples are maintained in a priority list, they are then analysed, and the semantic relation class is assigned based on votes, weighted by the similarity between the classes present on the kNN most similar examples. Tests showed, for instance, that the method is able to extract 10 different types of semantic relations, 8 of them corresponding to asymmetric relations, with an average score of 55.6%, as measured in terms of the F1 metric over Portuguese texts.

2.3 Opinion Mining and Sentiment Analysis

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis, and computational linguistics, to determine the attitude of a speaker or a writer with respect to some topic, or to determine the overall contextual polarity of a document. Previous work has addressed the problem of identifying the semantic polarity orientation of individual words, i.e. finding indications that specific words deviate from the norm in terms of encoding positive and negative sentiments [Lehrer, 1974]. Previous works have also showed that polarized words are good indicators of subjective sentences [Wiebe et al., 2001], and that automatically classifying words as either positive or negative can enable the automatic identification of the polarity of larger pieces of text, such as sentences or entire documents [Turney and Littman, 2003, Hassan and Radev, 2010].

For instance, Hatzivassiloglou and McKeown [1997] proposed a method to identify the polarity of adjectives based on conjunctions, linking the adjectives in a large corpus. They extract all conjunctions of adjectives from the corpus, and then they classify each conjunctive expression as either having the same or a different orientation. This procedure results is a graph, that the authors cluster into two different subsets of adjectives (i.e., the positive and the negative).

Turney and Littman [2003] used statistical measures of mutual information to find the degree of association between a given word and a set of positive/negative seed words. The authors collect co-occurrence statistics through search engine queries, consisting of the given word and one of the seed words that are known to be positive or negative, gathering the number of instances where the given word is physically close to the seed word in some document. From these results, the authors estimate the association towards the positive or the negative seed words, through the Pointwise Mutual Information (PMI) metric. In brief, we have that the PMI calculates the strength of the semantic association between

two words, w_1 and w_2, and is defined as follows:

$$\mathrm{PMI}(w_1, w_2) = \log_2 \left(\frac{\mathrm{p}(w_1 \wedge w_2)}{\mathrm{p}(w_1) \times \mathrm{p}(w_2)} \right) \tag{2.5}$$

In Formula 2.5, $\mathrm{p}(w_1 \wedge w_2)$ is the probability that w_1 and w_2 co-occur in a given textual window. If the words are statistically independent, the probability that they co-occur is given by the product $\mathrm{p}(w_1) \times \mathrm{p}(w_2)$. The ratio between $\mathrm{p}(w_1 \wedge w_2)$ and $\mathrm{p}(w_1) \times \mathrm{p}(w_2)$ is a measure of the degree of statistical dependence between the words. The log of the ratio corresponds to a form of correlation, which is positive when the words tend to co-occur, and negative when the presence of one word makes it likely that the other word is absent.

Takamura et al. [2005] instead proposed to use spin models to extract the semantic orientation of words. They construct a network of words using WordNet gloss definitions, thesaurus information, and co-occurrence statistics. The authors regard each word as an electron, with a given spin direction, and two neighbouring spins tend to have the same orientation from an energetic point of view. Their hypothesis is that since neighbouring electrons tend to have the same spin direction, neighboring words will tend to have similar polarity. The authors pose the task of defining spins as an optimization problem, and use the mean field method to find the best solution.

Kamps et al. [2004] construct a network based on WordNet synonyms, and then use the shortest paths between any given word and the seed words *good* and *bad* to determine the polarity. Hu and Liu [2004] also used WordNet to determine word polarity, but instead relying on synonyms and antonyms. For any word whose polarity is unknown, they search for it in WordNet, and check if any of the synonyms or antonyms of the given word has a known polarity. If so, they label the word with the label of its synonym, or the opposite label of its antonym.

Finally, Hassan and Radev [2010] used a random walk model defined over a word relatedness graph, to classify words as either positive or negative. These authors construct a graph where each node represents a word/part-of-speech pair, and where edges are used to connect nodes based on synonyms, hyperonyms, and similar-to relations from WordNet. For words that do not appear in WordNet, the authors use distributional similarity as a proxy for word relatedness. A list of seed words with a known polarity is used to label some of the nodes in the graph, and then a random walk process, where the seed words with a known polarity act as an absorbing boundary for the random walk, is used to assign the polarity classes to the unknown words.

Sentiment classification, or document-level sentiment classification, aims to find the general sentiment of the author in an opinionated text, classifying a document as expressing a positive or negative opinion, and assuming that the document is known to be opinionated. For example, given a product review, the task concerns with determining whether the reviewer is positive or negative about the product. Naturally, the same approach can also be applied to classify individual sentences.

Most existing techniques for document-level sentiment classification are based on supervised learning, although there are also some unsupervised methods. Sentiment classification can be formulated as a supervised learning problem with two class labels (positive and negative), which can be addressed

through maximum entropy modelling. The main challenge is to engineer a suitable set of features, which can be terms and their frequency (e.g., individual words or word n-grams, and their frequency counts), parts-of-speech tags (i.e., early studies have discovered that adjectives are important indicators of subjectivities and opinions), opinion words and phrases (e.g., *beautiful*, *wonderful*, *good*, and *amazing* are positive opinion words, and *bad*, *poor*, and *terrible* are negative opinion words), and negation words, which are particularly important because their appearance often changes the opinion orientation (e.g., the sentence *I do not like this camera* is negative, even though it contains the word *like*). Negation words must however be handled with care, because not all occurrences of such words actually mean a negation (e.g., the word *not* in *not only... but also* does not change the orientation direction). We also need to handle carefully words that in one context may be positive, but in another context may be negative. For example, the adjective *unpredictable* may have a negative orientation in a car review (e.g., *unpredictable steering*), but it can have a positive orientation in a movie review (e.g., *unpredictable plot*).

It is not hard to imagine that opinion words and phrases are the dominating indicators for sentiment classification. Thus, using unsupervised learning based on such words and phrases would be quite natural. Turney [2002] explained an algorithm for unsupervised learning which performs classification based on some fixed syntactic phrases that are likely to be used to express opinions. The algorithm consists of three steps:

1. Extract phrases containing adjectives or adverbs. The reason for doing this is that research has shown that adjectives and adverbs are good indicators of subjectivity and opinions. However, although an isolated adjective may indicate subjectivity, there may be an insufficient context to determine its opinion polarity. Therefore, the algorithm extracts two consecutive words, where one member of the pair is an adjective/adverb and the other is a context word. Two consecutive words are extracted if their POS tags conform to any of the patterns defined as interesting (e.g., two consecutive words are extracted if the first word is an adverb and the second word is an adjective, and the third word (which is not extracted) cannot be a noun);

2. Estimate the orientation of the extracted phrases using the pointwise mutual information towards known seed words (Formula 2.5);

3. Given a review, the algorithm computes the average opinion of all phrases in the review, and classifies the review as recommended if the average opinion is positive, and as not recommended otherwise.

Although classifying opinionated texts at the document level or at the sentence level is useful in many cases, these tasks do not provide the necessary detail needed for some other applications. A positive opinionated document on a particular object does not mean that the author has positive opinions on all aspects or features of the object. In a typical opinionated text, the author describes both positive and negative aspects of the object, although the general sentiment on the object may be positive or negative.

In order to identify the orientation of opinions expressed on an object feature in a sentence, Hu and Liu [2004] presented a lexicon-based approach that basically uses opinion words and phrases in

a sentence, to determine the orientation of the opinion, and that simultaneously handles negations and but-clauses in the sentence. Given the sentence *the picture quality of this camera is not great, but the battery life is long* as an example, the approach works as follows:

1. **Identifying opinion words and phrases:** Identify all opinion words and phrases in the sentence. Each positive word is assigned the opinion score of +1, each negative word is assigned the opinion score of -1, and each context dependent word is assigned the opinion score of 0. In the example above, after this step the sentence is turned into *the picture quality of this camera is not **great**[+1], but the battery life is **long**[0]*, because *great* is a positive opinion word and *long* is context dependent.

2. **Handling negations:** Negation words and phrases are used to revise the opinion scores obtained in the previous step. Thus, the sentence is turned into *the picture quality of this camera is not **great**[-1], but the battery life is **long**[0]*, due to the negation word *not*. It is important to notice that not every *not* means negation (e.g., *not only... but also*).

3. **But-clauses:** In English, *but* means contrary. A sentence containing *but* is handled by turning the opinion orientation before *but* and after *but* on the opposite. After this step, the above sentence is turned into *the picture quality of this camera is not **great**[-1], but the battery life is **long**[+1]*. As in the case of negation, not every *but* means contrary, (e.g., *not only... but also*).

4. **Aggregating opinions:** Finally, the resulting opinion scores are aggregated to determine the final orientation of the opinion on each object feature in the sentence. Let the sentence be s, which contains a set of object features $f_1, ..., f_m$ and a set of opinion words or phrases $op1, ..., op_n$ with their opinion scores obtained from steps 1, 2 and 3. The opinion orientation on each feature f_i in s is determined by the following opinion aggregation function:

$$\text{score}(f_i, s) = \sum_{op_j \in s} \frac{op_j.so}{\text{d}(op_j, f_i)} \tag{2.6}$$

   In Formula 2.6, op_j is an opinion word in s, $d(op_j, fi)$ is the distance between feature f_i and opinion word op_j in s, and $op_j.so$ is the orientation of op_i. If the final score is positive, then the opinion on feature f_i in s is positive. If the final score is negative, then the opinion on the feature is negative, and it is neutral otherwise.

Still regarding related work within the area of opinion mining, previous studies have also addressed the task of identifying the sources and targets of opinions, as well as identifying power relations from written dialogue. For instance, Peterson et al. [2011] analyzed formality in Enron email messages, exploring the factors that can affect the sender's choice of formality, relating it to social distance and power. Strzalkowski et al. [2010] explored language uses (e.g., notions of topic switching, as captured through lexical features, or linguistic expressions denoting lack of power, such as asking for approvals) that might indicate social constructs like leadership and power. Bramsen et al. [2011] classified messages accord-

ing to the direction of power (i.e., as being sent from a superior to a subordinate, or vice versa) through supervised learning.

2.4 Extracting Social Networks From Text

Previous work has also addressed the task of extracting networks of social relations between individuals from text. For instance, Elson et al. [2010] presented a method for extracting networks from nineteenth-century British novels and serials. The authors derive the networks from dialogue interactions, and thus their method depends on the ability to determine when two characters are in conversation. Their approach involves components for finding instances of quoted speech, attributing each quote to a character, and identifying when certain characters are in conversation. Finally, they construct a network where characters are vertices and edges signify an amount of bilateral conversation between those characters, with edge weights corresponding to the frequency and the length of their exchanges (i.e., the total exchanged information).

Merhav et al. [2012] addressed the task of automatically extracting information networks formed by recognizable entities as well as relations among them, from English social media sites. Their approach consists on using natural language processing tools (i.e., Stanford NER) to identify entities and extract sentences that relate such entities, followed by using text-clustering algorithms to identify the relations within the information network. The authors proposed a new term-weighting scheme that significantly improves results. We specifically have that Merhav et al. [2012] used the words that occur between the involved entities for extracting the relations, using a vector space model to represent each entity pair and the corresponding words between them, where the features were the unigrams, bigrams, and POS patterns (i.e., to+Verb, Verb+Prep, Noun+Prep, Verb and Noun) that occur in the sentence. The weights of these features are calculated using what the authors termed the domain frequency. The authors observed that TF-IDF just uses the frequency value of a term in a document, and does not consider the word frequency value in a relation. In TF-IDF, the weight of relational terms associated with some words (e.g., for the relation *married*, the terms *wife*, *spouse*, etc.) can have a huge value in any relation if a word appears several times in several documents, but instead some terms just make sense for some relations. The domain frequency is an approach that values a term according to its occurrence in each relation type, where the domain of a relationship is defined by the types of the entities in it (e.g., the domain between two persons is PER-PER).

To classify the relationships, the authors used an supervised clustering approach, namely the Hierarchical Agglomerative Clustering (HAC) algorithm, that consists on placing each entity pair in a distinct cluster, and then producing a hierarchy of clusters by sucessively merging clusters with the highest similarity, using a similarity threshold of 0.5. Finally, they label each cluster with a descriptive name that will represent all the relations within each cluster. For each cluster, they compute the arithmetic mean for each dimension of the feature vectors with basis on all the points in the cluster, and the feature with the largest mean value is selected as the cluster's label.

Evaluation tests showed that the domain frequency approach improves the accuracy, and unigrams+bigrams

are the best clustering features. The authors report on an average score of 85.6%, as measured in terms of the F1 metric.

Van De Camp and Van Den Bosch [2012] proposed machine learning-based tools for the classification of personal relationships in biographical texts, and for the induction of social networks from these classifications, effectively marking relations between pairs of persons as either positive, neutral, or unknown. The authors presented a case study based on several hundreds of biographies of notable persons in the Dutch social movement, where their classifiers marked relations between two persons (one of the persons, A, being the topic of a biographical article, and the other person, B, being mentioned in the article) determining whether the mention of person B can be labelled as positive or negative.

Lee et al. [2010] investigated the construction of social networks between people by extracting information from the Web. The authors, in a concrete and specific example, built networks that encode pairwise correlations between members of the 109th United States senate, as measured by co-occurrences in Google's search engine results, afterwards measuring node degree and strength as alternative approaches for estimating the importance of nodes, as well as the maximum relatedness sub network, its community structure, and its temporal evolution. The authors based their work on the idea that the co-occurrence of two people in many web pages implies that they are more closely related than two random counterparts. There are several advantages of using search engines to construct social relatedness networks. For instance, with a list of names, one can systematically count the number of web pages containing two names simultaneously, to assign the weights of association between all the possible pairs. This procedure enormously reduces the necessary efforts to extract social networks. Nonetheless, some errors may result from this procedure, for instance due to the fact that different people may share the same name. To avoid this situation, distinguishing words or phrases were added to all the search queries for each group, such as the words *senator* for US Senators, *physicist* for APS authors, and *baseball* for MLB players.

The Google correlation between two members of a group is defined as the number of pages returned using Google when querying for the pair of member names, and the additional distinguishing words. If no page is found for a pair, the pair is not considered to be connected. The constructed weighted networks are usually densely connected.

The degree and strength are basic quantities that estimate the importance of nodes in a weighted network. However, the weights on the links of two nodes with the same degree and strength are not necessarily identically distributed. In other words, just the number of links a node has (degree) and the sum of weights on the links the node has (strength) are not sufficient to fully conceive the node's character. Quantifying such different forms of weight distributions is important because it can distinguish whether a node's relationship with its neighbouring nodes is dominated only by a small portion of neighbours or if almost all the neighbours contribute similarly to the node's relationship.

The authors also conclude that the link density values of the Google correlation networks can be quite large, especially for the US Senate network, where almost every member is famous enough to appear on numerous webpages. Almost all the possible pairs of senators are connected. To avoid this, they suggest a new approach, called the Maximum Relatedness Subnetwork (MRS), where for each

node i, a directed link is connected from the node i to the other node j with which the node i has its maximum correlation value. It is possible for a node to have more than one directed link in the case of the multiple nodes with the same maximum correlation value. A MRS has the clear interpretation of consecutively connecting only the maximally related nodes.

Hassan et al. [2012] proposed a method to automatically construct signed social networks from online discussion posts, evaluating the extracted networks through social psychology theories of signed networks. The authors presented algorithms for identifying user attitude and for automatically constructing a signed social network representation. The resulting networks have a polarity associated with every edge. Edge polarity is a means for indicating a positive or negative affinity between two individuals. The proposed method was applied to a large set of online discussion posts.

The first step towards identifying attitude is to identify words with a positive/negative semantic orientation. Polarized words are very good indicators of subjective sentences, and their existence is highly correlated with the existence of attitude. The authors constructed a graph where each node represents a word/part-of-speech pair, connecting the nodes based on synonyms, hypernyms, and similar-to relations from WordNet. They then used a list of words with a known polarity to label some of the nodes in the graph. Then, they defined a random walk model where the set of nodes corresponds to the state space, and where the transition probabilities are estimated by normalizing edge weights. The walk continues until the surfer hits a word with a known polarity. If the absolute difference of the two mean hitting times (i.e., the mean hitting time from any word with unknown polarity to the set of positive seeds, and the set of negative seeds) is below a certain threshold, the word is classified as neutral.

In the next step, they examine different sentences to find out which sentences display an attitude from the text writer to the recipient, training a classifier to make this prediction. They used Support Vector Machines (SVM) for classification because this technique has been shown to be highly effective for traditional text classification. The features considered by the authors included unigrams and bigrams representing the words, parts-of-speech tags, and dependency relations connecting the two entities.

Finally, the authors built a network connecting participants based on their interactions, using the predictions made at the word and sentence levels to associate a sign to every edge. Unfortunately, the sign of an interaction cannot be trivially inferred from the polarity of sentences. For example, a single negative sentence written by A and directed to B does not mean that the interaction between A and B is negative. One way through which the authors attempted to solve this problem was by comparing the number of negative sentences to positive sentences in all posts between A and B, and classify the interaction according to the plurality value. Their experiments showed that this method does not perform well in predicting the sign of an interaction. Thus, they decided to pose the problem as a classical supervised learning problem, and trained a classifier using a set of features on a labled dataset. The features included numbers and percentages of positive/negative sentences per post, posts per thread, and so on.

In what concerns previous work within the area of social network analysis, most authors have focused on the analysis of networks that only encode positive interactions. For instance in online social networks, people tend to show only positive attitudes, by labelling others as friends. A few recent papers

have nonetheless taken the signs of edges into account. Brzozowski et al. [2008] studied the positive and negative relationships between users of Essembly, an ideological social network that distinguishes between ideological allies and nemeses, with the goal of predicting outcomes of group votes. Traditionally, online social network sites like Facebook and MySpace allow people to form links to friends but do little to qualify the semantic meaning of the friendship. As a result, many users collect friends on these sites, conflating acquaintances with friends. The authors hypothesize that improving the semantic granularity of friendship in a social network will increase the relevance of friend influences in filtering content. They examined user behaviour from Essembly, a fiercely non-partisan social network that allows members to post messages reflecting controversial opinions (e.g. *overall, free trade is good for American workers*). Essembly is unique in that it defines three semantically distinct but overlapping types of connections: friend, ally, and nemesis. In social networks, homophily is the principle that people tend to be connected with other people that are demographically and behaviourally similar to them. Two users who are in complete agreement have a similarity of 1, while two users who vote in complete opposition have a similarity of 0.

The author's experiments showed that friends and allies tend to be more ideologically similar. In short, people are more likely to agree with their allies than with their friends, and with their friends than with their nemeses. This provides strong evidence that the three semantically distinct networks capture different levels of ideological homophily. Tests also suggest that users are about 48% more likely to vote on a resolve if one or more of their friends voted on it. By contrast, if four or more friends voted on a resolve already, they are 66% more likely to vote on it. Curiously, in the cases where two or more nemeses voted on a resolve (but no friends or allies), the posterior drops to 38%.

Maniu et al. [2011] presented a study on a signed network derived from user interactions in Wikipedia articles. Their study addressed the principles underlying a signed network that is captured by social interaction, assessing connections with social theories such as structural balance and status. The signed network is built with basis on a local model for user relationships: for a given ordered pair of members (i.e., a generator and a recipient) of the online community, it will assign a positive or negative value, whenever such a value can be inferred. The authors approach aims at converting interactions into indicators of user affinity or compatibility: to give a brief intuition, deleting one text or reverting modifications (backtracking in the version thread) would support a negative link, while surface editing text or restoring a previous version would support a positive one. These modifications can be viewed as an interaction vector from a generator to a recipient that will form the basis for inferring signed edges between users. A positive or a negative interpretation is given to each of the atomic interactions (i.e., text insert, delete and replace, text reverts and restores). Then, for deciding a final link sign, for a given pair of contributors, the authors used a straightforward heuristic based on the text operations insert, delete and replace.

Kunegis et al. [2009] analyzed user relationships in the Slashdot technology news site, which allows users to tag other users as friends or foes, thus providing positive and negative endorsements. These authors computed global network properties, such as node centrality or the clustering coefficient, discussing particular properties of signed networks. They presented network analysis methods based on the concept of transitivity, which stipulates that relations between any two nodes in the network can be

described by paths between the two nodes. Multiplicative transitivity is motivated by the fact that triangles of users connected by an even number of negatively weighted edges can be considered balanced, which can be summarized by the phrase *the enemy of my enemy is my friend*.

As a measure of multiplicative transitivity, the authors proposed the signed clustering coefficient. The clustering coefficient is a characteristic number of a graph taking values between zero and one, denoting the tendency of the graph nodes to form small clusters. This coefficient is defined as the proportion of all incident edge pairs that are completed by a third edge to form a triangle. To extend the clustering coefficient to negative edges, the authors assume a multiplication rule for two incident signed edges that captures the intuition that *the enemy of my enemy is my friend* (i.e., an edge c completing two incident edges a and b to form a triangle must fulfil the equation $c = ab$). Therefore, the signed clustering coefficient denotes to what extent the graph exhibits multiplicative transitivity. The relative signed clustering coefficient is +1 when all triangles are oriented coherently. In networks with negative relative signed clustering coefficients, the sign multiplication rule does not hold.

Leskovec et al. [2010b] studied signed social networks generated from Slashdot, Epinions, and Wikipedia, also connecting their analysis to theories of signed networks from social psychology [Leskovec et al., 2010b,a]. The authors also came to the conclusion that the attitude of one user towards another can be estimated from evidence provided by their relationships with other members of the surrounding social network. For example, if A is known to dislike people that B likes, this may well provide evidence about A's attitude towards B.

In order to define the problem of edge sign prediction, the authors considered that given a full network with all but one of the edge signs visible, they should predict the sign of this single edge whose sign has been suppressed. They used a machine-learning approach to solve this problem, with a collection of features divided into two classes. The first class is based on the degrees of the nodes, which essentially record the aggregate local relations of a node to the rest of the world. In this class, they consider outgoing edges from a individual u and incoming edges to a individual v, and the total number of common neighbours of u and v in an undirected sense (i.e., the number of nodes that are linked by an edge in either direction with both u and v). The second class is based on the principle from social psychology in which it is possible to understand a relationship between individuals u and v through their joint relationships with third parties w (i.e., nodes that has an edge either to or from u and also an edge either to or from v). Each of these connections may provide different evidence about the sign of the edge from u to v, some favouring a negative sign and some favouring a positive sign.

The authors used a maximum entropy classifier to combine the evidence from these individual features into an edge sign prediction. The authors also used machine learning for deriving insights into the usage of these systems, based on the observed patterns of positive and negative edges. Specifically, maximum entropy models provide a coefficient associated with each feature, which suggests how the feature is being used by the model to provide weight for or against a positive edge sign. This provides a natural and appealing connection to classical theories from social psychology, which also offer proposals for how subsets of these features offer evidence for the sign of a hidden edge.

The more well-studied of the two main social-psychological theories is *structural balance theory*,

based on the common principles that *the friend of my friend is my friend*, *the enemy of my friend is my enemy*, *the friend of my enemy is my enemy*, and *the enemy of my enemy is my friend*. Concretely, if an individual w forms connections with the individuals u and v, then the structural balance theory posits that the edge $< u, v >$ should have the sign that causes the triangle on u, v, w to have an odd number of positive signs, regardless of edge direction.

In all experiments the authors reported the average accuracy and estimated maximum entropy coefficients over 10-fold cross validation. They note that the results are robust with respect to training dataset and evaluation metric. Generally, when using the full dataset rather than a balanced one, random guessing had and accuracy of approximately 80%. With the full dataset, the accuracy jumps to the 90-95% range and maintains roughly a 15% absolute improvement over random guessing.

2.5 Sentiment Slot Filling From Text

The sentiment slot filling task is a special case of the more general slot filling problem, as proposed in the TAC-KBP evaluation challenge [Min and Grishman, 2012]. In the slot filling task, we are given a list of entities, such as person and organization names, and we must locate information about these entities in a textual corpus in order to fill empty attributes describing the entity (i.e., the slots) with correct, non-redundant information. Slots can include, for instance, *per:employee_of* and *org:top_members/employees*. Some slots such as *per:date_of_birth* can only take on a single value, while others such as *per:member_of* can take multiple values [Min and Grishman, 2012].

In the specific case of the sentiment slot filling task, the slots indicate that the query entity has expressed either a positive or negative sentiment towards another entity. Alternatively, the slots may indicate that the entity is the target of either a positive or negative sentiment expressed by some entity.

For the purpose of the TAC-KBP sentiment slot filling task, only sentiments referring to KBP entity types (i.e., person, organizations or geo-political entities) were relevant, and only if they apply to an entity as a whole. Sentiments that scope over only one attribute of an entity were not relevant (e.g., *I love the Ravens* is relevant, while *The Ravens had a horrible season this year* is not). As document collection, the sentiment slot filling task at TAC-KBP used primarily blogs with discussion posts, although news articles could also be included.

Li et al. [2013] described the system that achieved the best results in the sentiment slot filling task at TAC-KBP 2013. For each query, they retrieved 150 related documents and parsed them using the Stanford CoreNLP system. They also divided the slots into 2 categories (i.e., positive and negative).

In the training set, they extracted the holder, target and the whole sentence according to the annotation, conducting also pronoun resolution, because sometimes the holder or the target can be a pronoun in the training data. In order to identify the sentiment holders and targets, the authors trained two maximum entropy models, specific for sequence classification (e.g., two CRF models) using four types of features. The four types of features that were considered are as follows:

1. Parts-of-Speech tags;

2. Whether the token is a named entity;

3. Whether the token is a subjective word;

4. Whether the token is a sentiment word.

As for identifying sentiment orientations, the authors used the dominant orientation of the opinion words in the sentence. The determination of sentence sentiment orientations was based on a general opinion lexicon. In order to extract the candidate fillers, the authors applied the following algorithm: for each sentence with a candidate, if its sentiment orientation satisfies to the query request and if its strength is bigger than a threshold, then the candidate is considered as the final filler.

Evaluation tests showed that the system obtained a mean precision of 8%, a recall of 17.8%, and a F_1 score of 11.1%.

2.6 Overview

This chapter presented the concepts necessary to understand the work that has been made in the context of my MSc thesis, together with the related work in which I based my work. Specifically, we have seen that:

- Proper representations for textual information are the starting point for any text analysis task. When we have a document collection to be used on some text analysis task, the first thing to do is splitting the individual documents into sentences, and then breaking up a each sentence into meaningful simpler units, such as words, through tokenization.

- Natural Language Processing explores how computers can be used to understand and manipulate natural language texts to do useful things. Typically, NLP is based on a pipeline of operations that involves tokenization, sentence splitting, parts-of-speech tagging, parsing, named entity recognition, and relation extraction.

- Parts-of-Speech tagging is the process of marking up the words in a text as corresponding to particular morphological classes, based on their definition and context. Through POS tagging, we can know the gramatical class (e.g. noun, verb, adjective, adverb, etc.) of each word.

- Named Entity Recognition is one of the most important sub-tasks of information extraction, where we can identify entity references, in textual documents, such as persons, locations and organizations. We can model Named Entity Recognition as a task of giving tags to words, much like parts-of-speech tagging.

- The relation extraction task is oftenly formulated as the detection and classification of semantic relationships within a sentence. Several authors have proposed machine learning techniques to address this problem, for instance formulating it as a binary supervised classification task defined on candidates to nominal expression pairs, where the candidates are then classified as being related or not related.

- Many problems in Natural Language Processing can be formulated as statistical classification problems. Such classifications can, for instance, be made through a maximum entropy model, a technique that offers a clean way to combine diverse pieces of contextual evidence in order to estimate the probability of a certain class occurring with a certain context.
- Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, and text analysis, to determine the attitude of a speaker or a writer with respect to some topic, or to determine the overall contextual polarity of a document. Previous works have also showed that automatically classifying words as either positive or negative can enable the automatic identification of the polarity of larger pieces of text, such as sentences or entire documents.

Chapter 3

Named Entity Recognition in Portuguese Texts

This chapter address the development of an efficient and robust Named Entity Recognition (NER) approach for the Portuguese language, using modern statistical models and relying on machine learning. In particular, and following on ideas from recent work in the area focusing on the English language, I addressed issues such as the representation of the text chunks corresponding to entities, the usage of features derived from large volumes of unlabeled text, the usage of external knowledge sources that can be incorporated as features, and how to effectively consider all these aspects within a practical NER system. The following section details the NER approach. Section 3.2 describes the obtained experimental results. Finally, Section 3.3 presents my conclusions together with a critical discussion.

3.1 An Entity Recognition System for Portuguese Texts

Named Entity Recognition concerns with the detection of entities (i.e., persons, organizations and locations) mentioned in textual documents. Named Entity Recognition (NER) is an important task in the context of Natural Language Processing (NLP). Automatic systems for NER accept as input a text segment, written in a particular language, and then identify, delimit and classify the expressions that refer to real-world entities, mentioned in the text. A typical categorization that is considered in NER systems involves the distinction of entities that correspond to persons, organizations, locations, and other entities. Persons are essentially textual expressions that refer to names of persons, with the option of considering the position or social *status* of the individual, if present (e.g., *President Cavaco Silva*). Organizations are expressions that refer to names of firms (e.g., *Banco Santander*) or political organizations (e.g., *United Nations*). The locations are entities related to specific geographical locations (e.g., *Iberian Peninsula*). Finally, a miscellaneous category is typically used for expressions that refer to entities that can not be classified according to previous types (e.g., *iPad*).

NER is an essential task for my work, namely because in order to perform relation extraction over a given textual document collection, the first step that needs to be taken is the recognition of the entities

contained in it.

Most previous studies have addressed NER tasks focused on texts written in English, although some developments have been reported in the context of applications to the Portuguese language. In this thesis, I developed a NER system for the Portuguese language, with the goal of improving the results obtained in previous works. I addressed the issue of developing an efficient and robust NER approach for the Portuguese language, using state-of-the-art (e.g., first-order or second-order statistical models for sequential data, based on the principle of maximum entropy and known in the literature as *Conditional Random Fields* (CRFs)), and also using machine learning. In particular, following the ideas of recent work with its focus on the English language [Ratinov and Roth, 2009], I addressed issues such as the representation of textual segments corresponding to entities (i.e., comparing the IOB and the SBIEO encodings for entities as individual word tags), the use of resources based on non annotated text (e.g., features derived from word clustering or derived from information related to occurrences of capitalized words), the external sources of knowledge that can be incorporated as features in the models (e.g., word lists and dictionaries) or the most effective way to consider all of these aspects in a NER system of practical use. In what regards features derived from information present in large volumes of non-annotated data, I used an hierarchical process that clusters words based on the contexts in which they occur, with the purpose of maximizing the mutual information of bi-grams, proposed by Brown et al. [1992]. The intuition behind the method related to a class-based language model, i.e. one where probabilities of words are based on the classes (clusters) of previous words, based on:

$$\mathrm{P}(w_1^N|C) = \prod_{i=1}^{N} \mathrm{P}(C(w_i)|\mathrm{C}(w_{i-1})) \times \mathrm{P}(w_i|C(w_i))$$

In this model there are two types of probabilities, including the probability of transition $\mathrm{P}(c|c')$ for a class c given its predecessor class c' and the probability $\mathrm{P}(w|c)$ for a word w in a certain class c. The probabilities can be estimated by counting the relative frequencies of unigrams and bi-grams. In order to determine the optimal classes C for a certain number of classes M, we can adopt a Maximum Likelihood approach to find assignments $C = \arg\max_C \mathrm{P}(W_1^N|C)$. A greedy agglomerative hierarchical clustering algorithm was proposed by Brown et al. [1992] in order to find the best clusters C. Each word is initially assigned to a specific group, and pairs of clusters are iteratively merged based on the criterion of mutual information, completing the process to achieve the number of groups M.

In order to use Brown clustering I used an open-source[1] implementation of the Brown procedure that follows the description given by Turian et al. [2010], together with two large collections of Portuguese texts, namely by using a set of phrases that combines the CINTIL corpus with news published in Público newspaper over a period of over 10 years, inducing thousand clusters of words, as done in previous related work [Ratinov and Roth, 2009, Turian et al., 2009].

Apart from global features obtained with methods of word clusters, I also held tests involving the introduction of a simple feature that essentially captures the probability of having each word appearing in the texts with the first letter capitalized. The values associated with this feature can be easily estimated

[1] `https://github.com/percyliang/brown-cluster`

Sentences	26329
Tokens	537064
Types	42957
Persons	9976
Localizations	5216
Organizations	6366
Miscelaneous	2844

Table 3.1: Statistical characterization for the NER dataset.

based on a set of unannotated texts (i.e., the same used in the induction of groups of words). Since entities typically correspond to words that often arise with the first letter capitalized in the text, this feature can provide a good indication of whether a type of particular *token* is or not used within entities mentioned.

In what regards to external sources of knowledge, I considered gazetteers with names of people, organizations and locations, as described in the Portuguese version of DBpedia. I also used lists of specific words corresponding to the most common first names or family names, collected from various sources on the Web (e.g., from Wikipedia pages showing lists popular of proper names or family names).

In the following subsections, I explain the use of CRF models to recognize the named entities, as well as the dataset considered to train and test CRF models. Finally, a separate subsection details the considered features.

3.1.1 The Considered Dataset

In this work, I trained and evaluated different NER models (i.e., first-order or second-order CRF models, using the IOB or SBIEO encodings, and using different sets of features) using the CINTIL[2] corpus of modern Portuguese, together with a cross-validation procedure which splits the data into 5 folds.

The CINTIL corpus presented some challenges in terms of annotation and representation. Specifically, the annotations in CINTIL consider two other types of entities, i.e. entities of type work (i.e., expressions denoting movies, books, paintings and similar works) or of the type event (i.e., competitions, conferences, workshops and similar events). As a preprocessing step, we converted the entities regarding to works and events into entities of the miscellaneous type. The tokenization model, used to split the texts into *tokens*, considered in the development of CINTIL, also expands common contractions (e.g., tokens like *daquela*, *aos*, e *nas* are respectively encoded as `de_ aquela`, `a_ os`, and `em_ as`), and separates the clitic pronouns of the corresponding verbs (e.g., the verb *ve-las* is encoded as `ve# -las`). As another preprocessing step, I used a set of regular expressions to replace the expansions made by the tokenization model used to create the CINTIL corpus, so that NER models trained using these data can be applied to Portuguese text processed with a different tokenization model that does not take any of these preprocessing decisions.

The software package used to perform named entity recognition was Stanford NER. The dataset had to be adapted in order to be encoded in the representation format accepted by Stanford NER, where

[2] `http://cintil.ul.pt/`

each line has an individual *token* along with the corresponding IOB or SBIEO tag.

The most common scheme for tagging words in the context of NER models is perhaps the IOB encoding, in which each *token* is marked with one of three labels, namely `O` that means `OUTSIDE`, `B` with the meaning of `BEGIN`, or `I` with the meaning of `INSIDE`. In addition, a suffix that indicates the type of the mentioned entity is attached to the labels `B` and `I`, (e.g., a suffix `PER` for people `ORG` for organizations, `LOC` for locations, and `MSC` for other entities).

Another popular tagging scheme is referred to as the SBIEO encoding, where *tokens* are marked with five labels, namely `O` with the meaning of `OUTSIDE`, `B` meaning `BEGIN`, `I` with the meaning `INSIDE`, `E` with the meaning `END`, and `S` meaning `SINGLE`. By separately indicating words that form one entity individually, or words that appear at the end of a segment corresponding to an entity, we can perhaps better use the sequential information in the classification task. Recently, Ratinov and Roth [2009] showed that NER models using the encoding scheme SBIEO usually outperform those that use the simpler IOB encoding.

The Stanford NER package already provides trained models for the English language, but lacks models for the Portuguese language. Nonetheless, it provides an easy framework to train new models with data from different sources. A NER model based on the formalism of Conditional Random Fields was trained using the part corresponding to the written CINTIL International Corpus of Portuguese, which is composed of 537064 annotated word tokens taken from texts collected from different sources and domains. Table 3.1 shows a statistical characterization for the corpora used in the evaluation experiments, after preprocessing the CINTIL corpus, presenting the number of *sentences* and the number of *tokens*, together with the number of entities of each type.

3.1.2 Using CRF Models for NER

The NER task is usually treated as a classification problem over sequential data, in which the objective is to automatically assign the most probable tag sequence $S =< s_1, s_2, \ldots, s_T >$ to a given sequence of observations $O =< o_1, o_2, \ldots, o_T >$ with length T. Accordingly, each sentence originating from a given text is treated as a sequence of words (i.e., the observed *tokens*), and the resulting sequence of tags (i.e., the resulting annotations, which classify specific segments of text) encode the entities mentioned in the sentence.

Regarding the statistical models that support sequential classification, we have that an approach based on the principle of maximum entropy, known in the literature as *Conditional Random Fields* (CRFs), is nowadays oftenly used. This approach involves the calculation of the conditional probability for each sequence of output labels, given a sequence of input *tokens* [Lafferty et al., 2001]. In CRF models, the conditional probability of a sequence of labels S, given a sequence of it tokens O, is given by the following equation:

$$\mathrm{P}(S|O) = \frac{1}{Z_0} \exp\left(\sum_{t=1}^{T}\sum_{k} \lambda_k \times \mathrm{f_k}(s_{t-2}, s_{t-1}, s_t, O, t)\right)$$

In the above formula, each $\mathrm{f_k}(s_{t-2}, s_{t-1}, s_t, o, t)$ is a feature function. The features are typically binary

Persons	66718
Organizations	20696
Localizations	131283
First Names	7421
Last Names	8124

Table 3.2: Number of references in the lists of names and gazetters that are considered.

and a feature function returns the value zero for all cases except if s_{t-2}, s_{t-1} and s_t take some particular labels, and if the observations also possess certain properties. It should be noted that the feature functions in CRF models typically use a small number of prior positions in the sequence of labels (e.g., the previous two positions, represented by s_{t-2} and s_{t-1}) and in the sequence of *tokens*, so that models can be used efficiently. The weights λ_k associated with each feature function are learned automatically through training with supervision (i.e., based on annotated data). In order to have a conditional probability with a value between 0 and 1, we calculate the normalization factor Z_0 considering all the sequences and all possible labels:

$$\mathrm{Z}_0 = \sum_{s} \exp\left(\sum_{t=1}^{T}\sum_{k} \lambda_k \times f_k(s_{t-2}, s_{t-1}, s_t, o, t)\right)$$

To train a CRF model, the main function, given the sequences of observed *tokens*, involves a likelihood of sequences of labels, corresponding to:

$$\mathrm{L} = \sum_{i=1}^{N} \log(\mathrm{P}(s^{(i)}|o^{(i)})) - \sum_{k} \frac{\lambda_k^2}{2\sigma^2}$$

In the formula, $\{< o^{(i)}, s^{(i)} >\}$ refers to the set of N annotated training sequences. The second sum corresponds to a Gaussian *prior* on the parameters, of zero mean and variance σ^2, which facilitates the optimization process because it makes the surface of the likelihood function become strictly convex. Typically, the parameters λ_k are adjusted, in order to maximize the penalized log likelihood, using a quasi-Newton method known as the L-BFGS algorithm [Malouf, 2002].

After training a classification model, one can then use it to assign the most likely sequence of labels for new *token* sequences (i.e., *decode* the text). Since CRF classifiers produce results in the form of probabilities, which can also be translated into costs, a common approach involves the use of dynamic programming (e.g., the Viterbi algorithm) for assigning the label sequence that maximizes the overall probability on the sequence of words, given the consistency of the solution.

In this work, I used an existing NER system that uses CRF models and does *decoding* of new texts based on the Viterbi algorithm, namely the Stanford NER system.

3.1.3 The Considered Features

Stanford NER uses linear chain Conditional Random Field (CRF) sequence models, which predict sequences of labels for sequences of input samples (i.e., sequences of words), coupled with feature ex-

tractors for named entity recognition. The considered features are as follows:

- **Words:** The current words, previous words, and next word, within a window of two tokens, are all used as features in the NER models.
- **Previous tags:** Tag of the previous word or tags of the two previous words, in the case of second-order models.
- **Word Shape:** A shape function that categorizes the current word in one of 24 possible classes (e.g., MIXEDCASE, ALLCAPS, NUMBER, etc.), was used to create features that combine the shape and the words from the current, the previous and the next positions (e.g., previous word + current shape, or current word + next shape).
- **Sufixes and Prefixes:** Prefixes and suffixes of the current word.
- **First in Sentence:** The current word appears in the first position of the sentence. This parameter has value 1 if the condition is true and 0 otherwise.
- **Upper-cased:** The first letter of the current word is upper-cased. This parameter has value 1 if the condition is true and 0 otherwise.
- **Names List:** The presence of the current word in predefined lists of names was also used as a feature, where the lists have only one word per line, which is case insensitive. I used two lists in the training of these models, namely lists containing common person first names and common family names - see Table 3.2.
- **Gazetteers:** Gazetteers are similar to the lists of names, but they may have names with multiple words, and the names are always associated with a class (e.g. *United Kingdom* - `LOCATION`). These features check if the current word is a part of any name in a gazetteer. I used a gazetteer containing a complete set of names for each one of the considered types (i.e., Portuguese locations, Portuguese organizations, and Portuguese Persons). These gazetteers are built using the DBPedia dataset, that contains structured information created with basis on Wikipedia. Table 3.2 shows the size of the three gazetteers that were used.
- **Brown clusters:** This feature, correspond to the cluster (i.e., group of words) identifier of the current word.

3.2 Experimental Results

This chapter describes the experimental validation performed on the named entity recognition model created for the Portuguese language.

I used a five-fold cross-validation methodology. In k-fold cross-validation the data is partitioned into k equal size samples. One of this k samples is left out of the training process (i.e., is retained to perform the validation), and the remaining $k-1$ samples are used to train a NER model. This process is repeated,

		Evaluation based on Entity Spans																Aval based on Tokens			
		PER			LOC			ORG			MISC			ALL							
Model	Characteristics	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	A	P	R	F_1	A
CRF 1º Order	Basic	81.77	53.70	64.53	75.75	56.09	63.99	61.28	53.31	53.19	70.67	31.70	43.48	77.35	51.89	62.05	77.38	78.04	56.66	65.54	96.05
IOB	+Word Clusters	81.83	53.80	64.64	75.90	56.00	64.00	60.96	53.65	53.40	70.73	31.61	43.39	77.34	52.01	62.12	77.36	78.00	56.70	65.55	96.05
	+Gazettes	83.26	64.39	72.51	70.89	69.88	70.01	62.27	53.96	55.39	70.65	**34.02**	**45.70**	77.41	**60.76**	**68.06**	77.43	78.67	**63.98**	**70.52**	**96.62**
	+Lowercase	83.26	64.39	72.51	70.89	69.88	70.01	62.27	53.96	55.39	70.65	**34.02**	**45.70**	77.41	**60.76**	**68.06**	77.43	78.67	**63.98**	**70.52**	**96.62**
CRF 2º Order	Basic	79.49	52.04	62.52	72.91	54.78	62.00	**74.47**	54.85	**63.05**	68.17	32.98	44.39	75.24	50.48	60.34	75.27	76.50	56.05	64.56	95.48
IOB	+Word Clusters	81.77	53.66	64.50	75.74	56.27	64.11	61.16	53.32	53.27	70.33	31.57	43.27	77.33	51.87	62.02	77.35	78.03	56.61	65.50	96.05
	+Gazettes	83.33	64.31	72.49	71.23	70.02	**70.21**	62.00	52.50	54.86	70.81	33.74	45.45	77.54	60.66	68.05	77.57	**78.79**	63.85	70.50	**96.62**
	+Lowercase	83.33	64.31	72.49	71.23	70.02	**70.21**	62.00	52.50	54.86	70.81	33.74	45.45	77.54	60.66	68.05	77.57	**78.79**	63.85	70.50	**96.62**
CRF 1º Order	Basic	83.39	54.04	65.30	76.59	56.14	64.35	60.30	53.53	52.66	**72.54**	30.80	43.00	77.93	51.92	62.26	77.95	77.18	55.05	64.15	95.98
SBIEO	+Word Clusters	83.37	54.04	65.30	76.63	56.14	64.37	60.43	53.67	52.77	72.15	30.91	43.02	77.93	51.97	62.30	77.95	77.13	55.13	64.19	95.98
	+Gazettes	84.19	**64.55**	**72.99**	70.35	70.27	70.00	61.85	53.92	55.02	72.07	33.27	45.30	77.68	60.53	68.02	77.71	77.19	62.28	68.90	96.54
	+Lowercase	84.19	**64.55**	**72.99**	70.35	70.27	70.00	61.85	53.92	55.02	72.07	33.27	45.30	77.68	60.53	68.02	77.71	77.19	62.28	68.90	96.54
CRF 2º Order	Basic	83.52	53.72	65.13	**76.74**	56.04	64.31	60.55	53.81	53.11	71.60	30.68	42.72	**78.13**	51.86	62.27	**78.14**	77.39	54.86	64.09	95.97
SBIEO	+Word Clusters	83.18	53.87	65.10	76.55	56.33	64.43	60.53	53.37	52.74	71.92	30.85	42.98	77.91	51.83	62.17	77.92	77.14	54.68	63.89	95.96
	+Gazettes	**84.43**	64.48	73.05	70.20	**70.28**	69.95	61.99	**54.96**	55.17	71.53	33.25	45.14	77.70	60.47	67.99	77.72	77.08	62.29	68.86	96.53
	+Lowercase	**84.43**	64.48	73.05	70.20	**70.28**	69.95	61.99	**54.96**	55.17	71.53	33.25	45.14	77.70	60.47	67.99	77.72	77.08	62.29	68.86	96.53

Table 3.3: Results obtained with the CINTIL corpus, using cross validation with 5 folds.

using all k parts as the validation sample. In the end, an average score is computed over the k results, producing a single estimation that represents the quality of the final model.

In this thesis, I evaluate the NER experiments, with texts written in Portuguese, through some of the common metrics that have been used in previous works in the area. Specifically, I used precision, recall, the F_1 measure, and accuracy.

Precision is the fraction of discovered associations that are correct, while recall is the fraction of relevant associations that are discovered. In simple terms, high recall means that the NER model recognised most of the entities present in the dataset, while precision reveals if most of the entities recognised are correct. The accuracy is the overall correctness of the system, i.e the tags that the system got right divided by all the total number of tags assigned. Finally, the F_1 measure considers both precision and the recall to compute an overall score. The F_1 measure score can be interpreted as an harmonic average of the precision and recall. The evaluation was measured based of tokens and spans, i.e. the metrics were calculated using each word individually and using entities (e.g., a entity *United Kingdom* have 2 tokens and corresponds to one span).

The results are presented in terms of the various evaluation metrics, and in terms of the quality of results when assigning the IOB or SBIEO tags (i.e., accuracy of individual predictions as well as micro-average precision, recall and F_1 scores), as well as when classifying spans corresponding to named entities (i.e., the precision, recall and F_1 score, for each entity type, again terms of micro-averaged scores). Since the tests with the CINTIL corpus were made by means of cross-validation, we report the average values calculated with basis on all 5 folds. Table 3.3 presents the mean values—– calculated with basis on the results obtained in each one of the 5 folds, using cross-validation with the CINTIL corpus.

The obtained results contradict previous works where the results reported by Ratinov and Roth [2009] in the case of NER for texts in English defend that the SBIEO encoding consistently outperforms the results obtained with the IOB encoding. The inclusion of additional features, particularly features using external knowledge sources, do not increased the performance in some cases.

After manually analyzing the errors produced by different NER models, I could establish that a major source of error lies in the distinction between the types organization and location for specific entities. The most common case occurs in references to countries (for example, an entity such as `Spain / ORG-S`, which can be seen as representing a specific geographical location or a political organization). This kind of ambiguity in the semantic meaning of the entities are not easily resolvable through automatic systems

Model	Characteristics	Model size	Time to annotate 10K Tokens
CRF 1º Order	Basic	106MB	00m01s
IOB	+Word Clusters	105MB	00m01s
	+Gazetteers	112MB	00m08s
	+Lowercase	112MB	00m08s
CRF 2º Order	Basic	105MB	00m09s
IOB	+Word Clusters	106MB	00m10s
	+Gazetteers	111MB	00m17s
	+Lowercase	111MB	00m17s
CRF 1º Order	Basic	177MB	00m05s
SBIEO	+Word Clusters	177MB	00m05s
	+Gazetteers	183MB	00m12s
	+Lowercase	183MB	00m12s
CRF 2º Order	Basic	175MB	01m32s
SBIEO	+Word Clusters	176MB	01m30s
	+Gazetteers	184MB	01m39s
	+Lowercase	255MB	01m39s

Table 3.4: Computacional performance for the different models.

for named entity recognition.

NER models were also evaluated in terms of efficiency and computational performance, comparing the size of the resulting models (in MBytes) and the time (in seconds) required to process 10,000 words over a modern computer system. Table 3.4 presents the results, showing that the gains in term of quality typically involve much larger models, although the time required to process new documents remains relatively unchanged when adding new features to the models. However, the use of second-order CRF models, or of the SBIEO encoding, is significantly slower, perhaps not compensating the quality gains of the results.

3.3 Conclusions and Critical Discussion

In this section, I revisited the development of NER systems for the Portuguese language, using modern statistical models in conjunction with machine learning. In particular, I compared different models and different representations for the mentioned entities, and quantified the impact of considering different characteristics derived from large volumes of non annotated text or external sources of knowledge. My results showed that first-order CRF models, using the IOB coding, achieved the best results in terms of the quality of assignments. The inclusion of additional features, particularly features using external knowledge sources, do not increased the performance in some cases, quite the contrary. In fact, the overall results showed that IOD encoding is better than the SBIEO encoding, leaving the hypothesis that something went wrong, possibly the word clusters used were generated by a noisy dataset or even the gazetteers used may contain several entities that do not correspond really to entities.

For future work, it would be interesting to use this method with different external knowledge sources and different features, in order to discover the reason of the results obtained. It would be interesting to also experiment with NER on texts of more informal domains (e.g., in social Web content, such as messages from the Twitter and Facebook services). In the context of informal domains, experimenting

the use of transfer learning methods is also particularly interesting, combining small amounts of training data consisting of documents in the target domain (e.g., a small number of messages from service like Twitter, where their are entities annotated) with sets of existing data on similar domains (e.g., sets of journalistic texts).

Chapter 4

Extracting Signed Networks From Text

This chapter describes a relation extraction system, which performs the extraction of support and opposition relations between persons mentioned in documents written in the Portuguese language.

Leveraging the extracted relations, I also present a method for extracting signed social networks from newswire documents, which can subsequently support many different types of analysis. I propose to derive the networks from meaningful co-occurrences of person names within individual sentences, classifying the co-occurrences according to their semantic opinion polarity orientation. This classification is particularly important, given that the relations between individuals, as expressed on newswire documents, often reflect a mixture of positive (i.e., friendly) and negative (i.e., antagonistic) interactions involving opinion holders and targets, and simple co-occurrences by themselves do not hold a significant meaning. I used a bootstrapping method for performing the extraction and classification, adapting the previously proposed Snowball approach to my particular problem domain. I report on large-scale extractions from newswire documents related to politics, written in Portuguese.

The following section presents the problem and gives an overview of the relation extraction system. Section 4.2 describes an entity disambiguation approach used to group the entity names that correspond to the same real world entity. Section 4.3 presents the relation extraction system in detail. Finally, Section 4.5 presents the experimental results that were obtained. Finally, Section 4.6 presents the conclusions about this particular experiment.

4.1 Introduction

The social sciences strive to understand the political, social, and cultural world around us, but they are generally impaired by limited access to the quantitative data sources that are enjoyed by most of the hard sciences. However, the analysis of quantitative information derived from document streams, such as newswire texts or social media contents, holds an enormous potential to solve problems in a variety of social science disciplines, through massive data analysis [Zhu, 2010]. Political science, in particular, can stand to benefit significantly from massive analysis of newswire documents, as the field is primarily concerned with current events involving entities that are widely covered in the media. Previous

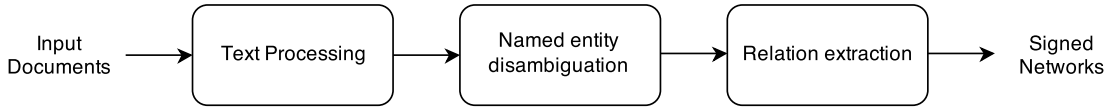

Figure 4.1: Pipeline of operations in the proposed approach.

work in the area has, for instance, addressed the development of scalable systems that can provide instant access, for students and scholars in areas such as political science, sociology, or marketing, to interesting statistics on millions of entities mentioned across a different types of Web corpora (e.g., newspapers, blogs, patent records, legal documents, etc.) [Bautin et al., 2010, Key et al., 2010], as well as text-driven forecasting tasks, such as predicting opinions or political orientation from textual documents [Zhu, 2010, Smith, 2010].

On the other hand, social network analysis has long been a hot research topic within several different disciplines, with several pivotal results such as the 6-degrees of separation measured by Milgram [1967], the small world network phenomena observed by Watts and Strogatz [1998], or the methods for finding important central nodes introduced by Freeman [1977], just to name a few. Compared to other quantitative approaches (e.g., conventional regression analysis, with basis on data inferred from textual documents [Smith, 2010]) which focus on individual instances and their attributes, social network analysis leverages on relations between instances, effectively supporting the analysis of the interdependence and flows of influence among individuals and groups.

In this chapter, I present a method for extracting signed social networks from newswire documents, which can subsequently support many different types of analysis that are relevant to the political sciences, such as the discovery of meaningful communities (i.e., sets of political actors that interact more between themselves than with other actors, or sets of political actors where both support within-group relations and opposition between-group relations are dense [Yang et al., 2007, Doreian and Mrvar, 2009]), the estimation of political influence (e.g., discovering the political actors that are more central in the network, of the actors that aggregate more friendly or antagonistic opinions), or the prediction of the polarity orientation for specific associations between political actors. I propose to derive the networks from meaningful co-occurrences of person names within individual sentences (i.e., co-occurrences that are associated to specific linguistic patterns, commonly used to encode semantic relations), and I also propose to classify the co-occurrences according to their semantic polarity orientation. These classifications are particularly important, given that the relations between individuals, as expressed on newswire documents, often reflect a mixture of support (i.e., friendly) and opposition (i.e., antagonistic) interactions from holders to targets, and simple co-occurrences by themselves do not hold a significant meaning. A bootstrapping method was used for performing the extraction of relations between individuals, through an adapted version of the previously proposed Snowball approach, classifying these relations as encoding *support* or *opposition* between pairs of individuals mentioned in the news.

The prototype system developed in the context of my MSc thesis is composed by three major modules, as shown in Figure 4.1. The system receives as input a documents collection, which is composed

by the a set of news that will be used to build the network. This input goes then through the text processing module, where each document is split into sentences, and the named entities are recognized from each individual sentence. The named entities recognised in the text are then disambiguated through a simple procedure that joins together co-referential entity references, in the named entity disambiguation module. I apply some heuristic filters on the entities recognised in the document collection, in order to eliminate some undesired candidate names from the results of the recognition stage, and then group the remaining entities into several clusters according to their similarity with each other. Each one of the clusters represents, in practical terms, a real world entity that is referenced through several names in the documents. The name that is chosen for each real world entity is the biggest name in each cluster, which means that all the entities inside that cluster have a new name assigned. The names of the entities are updated in the document collection and the sentences are used as input for the relation extraction module.

In order to extract the relations, I propose to use a bootstrapping approach that receives the document collection previously processed by the disambiguation module and two sets of user-provided seeds of entity pairs (i.e., one set representing examples of entities with the relation that we want extract, and one set with the opposite relation), and automatically generates and evaluates patterns for new pairs with the same relation.

In the results from the relation extraction, we can have multiple relations between the same pair of entities. In this case, we aggregate all the relations corresponding to the same entity pair, assuming that the polarity of the relation is the most frequent polarity in all the relations (e.g., if the same pair of entities has three support relations and one opposition relation, we assume that it has a support relation). A signed graph is finally built from the candidates, by creating an arc for each aggregate from the previous step, giving it a sign according to the classification that was made for the aggregate. Optionally, the graph can be filtered so as to only consider edges whose support (i.e., the number of candidates in the aggregate) is above a given threshold, or whose semantic orientation score is above a given confidence. The system finally returns the signed network, where the vertices are the named entities and the links are the relations between them.

In the following sections, the main approaches taken in each module are detailed.

In order to test the quality of the system, I report on a case study involving large-scale extractions from newswire documents related to politics, namely documents published on the *Público* online newspaper, from the year 2009 up to the year of 2013. To evaluate the proposed method for social network extraction, I relied on two different ground-truth datasets that were built automatically with basis on politicians that once sat in the Portuguese national parliament and that appear in the documents considered, together with their respective party affiliations. Using this information we can consider individuals with the same affiliation to hold support relationships between themselves, and individuals with different affiliations to hold opposition relationships.

4.2 Disambiguating Person References

The disambiguation of candidate person names is particularly important for generating results with high quality, given that important political actors are often referred to in the media through a variety of different names (e.g., in 2009, the Portuguese prime minister José Socrates was often referred to as just *Socrates*, or through the nickname *Pinocrates*).

Initially, we start by finding all the person names present in the document collection, applying a named entity recognition model in each sentence of each document. After this, we analyse each name and assign it the respective gender, based on two lists of names:

1. A list with 1717 unique female first names;
2. A list containing 1567 unique male first names;

If some entity name has a reference to a name in those lists, we assume that the entity is either male or female, and otherwise I assume that the entity is unknown. Considering a large collection of documents, where many entities occur, the named entity disambiguation process possibly will take several weeks or months. In order to avoid this, I propose to pre-filter the candidate set of names, by applying some heuristic filters. The actual set of heuristics is as follows:

- Discard all the names that just appear once in the entire document collection, removing also the sentences where they appear, in order to avoid them in the entity relation extraction steps. When a name with an unknown gender is disambiguated, we compare this name with all the other names from the entire document collection. Considering a large set of entities, comparing one entity name with all the others can take several hours or even days. Taking into account that the dataset from *Público* has thousands of entity names, I decide to remove the entities that only appear once.
- Remove all the status and work positions prior to the person names inside the entity. Entities containing the same work positions can have a high similarity even if the persons in those entities are completely different (e.g., *prime minister Passos* and *prime minister Sócrates* are two similar strings). In order to address this problem, I created a list with 4417 different status terms and work positions that I used to filter the entity names and prepare them for the disambiguation (e.g., if we have *prime minister Passos* and *prime minister Sócrates*, the algorithm compares *Passos* and *Sócrates*). As a consequence of this heuristic filter, we also discard entity names that only represent work positions (e.g., *prime minister*, *secretary*, etc.). In other words, we discard an entity if it does not contain a first name and if it contains a work position.

In order to perform the disambiguation, I divide all the names in three lists (i.e., male names, female names, unknown names), depending on the respective gender, and I then apply the Jaro-Winkler TF-IDF similarity with a threshold value of 0.85 to compare and cluster together all person names having a similarity of 0.85 between themselves, based on the following rules:

- Compare each male name with all the other male names in the list;

- Compare each female name with all the names present in the female list;
- Compare each unknown name in the unknown names list with all the names in the document collection. This requires a great effort and takes a long time, depending on the document collection size. All the other heuristics are therefore important, in order to avoid that this case occurs several times.

In the next step, we compute the transitive closure of the matches. A relation R on a set X is transitive if, for all x, y, z in X, whenever xRy and yRz then xRz. Symbolically, this can be denoted as: if $x \rightarrow y$ and $y \rightarrow z$ then $x \rightarrow z$. More precisely, if an entity x is similar to y and y is similar to z, then x is similar to z. With basis on this, we merge together the clusters of entities where these rules are verified. Finally, we use the biggest name in each cluster as the name of the real world entity to which every name in that cluster corresponds. The system iterates through all the sentences of the document collection, and replaces the ambiguous named entities by the disambiguated entity name.

The considered string similarity metric is based on the Jaro-Winkler similarity between the two candidate names. Jaro-Winkler is a variant of the Jaro string similarity measure, using a prefix scaling parameter, in order to favour strings that share a prefix of a determined length:

$$d_w = d_j + (\ell p(1 - d_j)) \tag{4.1}$$

In the previous formula, l is the length of the common prefix up to a maximum of 4 characters, and p is scaling factor to determine how the score is adjusted when having common prefixes. The parameter d_j corresponds to the value of the original Jaro similarity metric, which corresponds to the formula shown bellow, where m equals the number of matching tokens, and t corresponds to half of the number of matching tokens which appear in a different sequence order. The Jaro distance d_j between two given strings s_1 and s_2 is give by:

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{4.2}$$

The Jaro-Winkler TF-IDF is a hybrid approach between the Jaro-Winkler measure and the TF-IDF similarity scheme, in order to use soft token-matching with the JaroWinkler distance metric. TFIDF, which is widely used in the information retrieval community, can be defined as:

$$\mathrm{TFIDF}(\mathrm{S}, \mathrm{T}) = \sum_{w \in \mathrm{S} \cap \mathrm{T}} log(\mathrm{TF}_w, \mathrm{S} + 1) \times log(\mathrm{IDF}_w) \tag{4.3}$$

In the formula, TF_w, S is the frequency of word w in S, and IDF_w is the inverse of the fraction of names in the corpus that contain w.

Jaro-Winkler TF-IDF is then the highest TF-IDF token-based string similarity between the candidate name tokens and the other name, using the Jaro-Winkler distance with a specific threshold when matching the individual tokens. If CLOSE(θ, S, T) is the set of words $w \in S$ such that some $v \in T$ exists where $\mathrm{JaroWinkler}(w, v) > \theta$, and for $w \in$ CLOSE(θ, S,T), let $\mathrm{N}(w, T) = \max_{v \in T} \mathrm{JaroWinkler}(w, v)$. We define

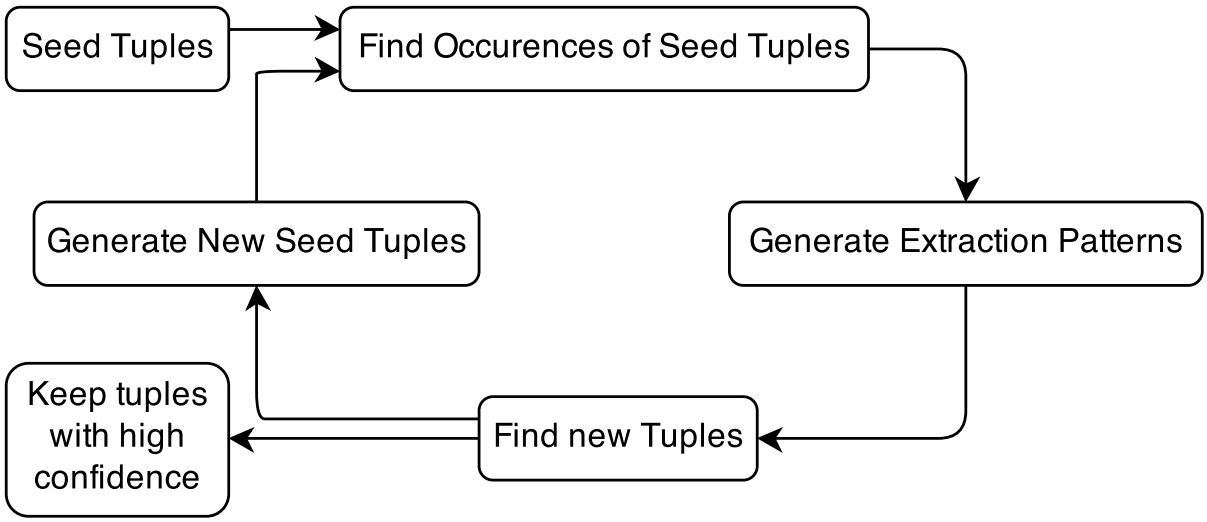

Figure 4.2: The architecture of Snowball, a partially-supervised information extraction system.

the final metric as:

$$\mathrm{JaroWinklerTFIDF(S,T)} = \sum_{w \in \mathrm{CLOSE}(\theta,S,T)} \log(\mathrm{TF}_w, \mathrm{S}+1) \times \log(\mathrm{IDF}_w) \times \mathrm{N}(w, \mathrm{T}) \qquad (4.4)$$

4.3 Adapting Snowball for the Extraction of Support and Opposition Relations

A bootstrapping method was used for performing the extraction of relations between individuals, through an adapted version of the previously proposed Snowball approach [Agichtein and Gravano, 2000]. Snowball starts with a small set of user-provided seed tuples for the relation of interest (i.e., example tuples for support or opposition relations), and automatically generates and evaluates patterns for extracting new tuples with the same relation. Snowball also uses a strategy for evaluating the quality of the patterns and the tuples that are generated in each iteration of the extraction process, where only those patterns that are regarded as being sufficiently reliable will be kept for the following iterations of the algorithm - see Figure 4.2.

In my extension of Snowball, the system starts with two user-provided sets of seed tuples, one for the relation of interest (i.e., example tuples for support or opposition relations), and another for the oposite relation of interest (i.e., example tuples for the oposite relation that we want to extract).

Snowball is initially given the example tuples, as seeds. For every such person-person tuple $<$ PER, PER $>$, Snowball finds segments of text in the document collection where two specific person entities occur close to each other, using the named entity tagger explained in Chapter 3 to recognise these entities. Once the entities in the text documents are recognized, Snowball can ignore unwanted entities (e.g., LOCATIONs and ORGANIZATIONs), focus on occurrences of PERSON entities, and analyze the context that surrounds each pair of such entities to check if they are connected by the right words. As a first step, Snowball uses just the sentences containing tuples $<$ PER, PER $>$, where both persons

where given initially as a seed tuple.

A crucial step in Snowball is the generation of patterns to find new tuples in the documents. In order to generate a pattern, Snowball groups occurrences of known tuples in documents, if the contexts surrounding the tuples are similar enough. More precisely, Snowball generates a tuple for each string where a seed tuple occurs, and then clusters these tuples. Snowball uses the sentences containing tuples, where both persons are in the set of seed tuples, and analyzes the text that connects the entities e_1 and e_2 to generate patterns. In order to represent each one of these text segments, Snowball keeps a 5-tuple containing the context before the first entity (left context), the name of entity one (e_1), the context between the two entities (middle context), the name of the second entity (e_2), and the context after the second entity (right context). The algorithm represents the left, middle, and right contexts associated with an extraction pattern as vectors of weighted terms calculated with TF-IDF. These weighted term vectors represent the respective context as a feature vector where the features are the words and also Brown word clusters, that were explained in Section 3.1.2.

The degree of match Match(t_p, t_s) between two 5-tuples $t_p =< l_p, t_1, m_p, t_2, r_p >$ (with entity types t_1 and t_2) and $t_s =< l_s, t'_1, m_s, t'_2, r_s >$ (with entity types t'_1 and t'_2) is defined as:

$$\mathrm{Match}(t_p, t_s) = \begin{cases} l_p \times l_s + m_p \times m_s + r_p \times r_s & \text{if the entity types are equal} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

In order to generate a pattern, Snowball clusters the tuples using a simple singlepass clustering algorithm [Day et al., 1997], using the Match function to compute the similarity between the vectors with a minimum similarity threshold τ_{sim}. The *left* vectors in the 5-tuples of clusters are represented by a centroid $\overline{l}_s$. Similarly, we collapse the middle and right vectors into $\overline{m}_s$ and $\overline{r}_s$, respectively. These three centroids, together with the original entity types (which are the same for all the 5-tuples in the cluster), form a Snowball pattern $< \overline{l}_s, t_1, \overline{m}_s, t_2, \overline{r}_s >$.

In this work, the patterns are generated using a different clustering approach of that from the original Snowball algorithm, based on DBScan [Ester et al., 1996]. DBScan requires two parameters, namely (1) a threshold similarity (i.e., DBScan_eps), and (2) the minimum number of instances required to form a cluster (i.e., DBScan_min_points). DBScan starts with an arbitrary tuple that has not been visited. This tuple's neighbours are retrieved, and if this set contains a sufficient amount of tuples, a cluster is started. Otherwise, the tuple is labeled as noise. If a cluster has been started, then the neighbours of the previous tuple's neighbours are retrieved and analized as before. The process continues, until the cluster is completely found. Then, a new unvisited tuple is retrieved and processed, leading to the discovery of other clusters.

After generating patterns, Snowball scans the collection to discover new tuples, by matching text segments with the most similar pattern, if any. Each candidate tuple will then have a number of patterns that helped generate it, each with an associated degree of match. Snowball uses this information, together with information about the selectivity of the patterns, to decide what candidate tuples should actually be added to the set of final tuples.

In order to estimate the quality of the patterns, we can weight them based on their selectivity, and

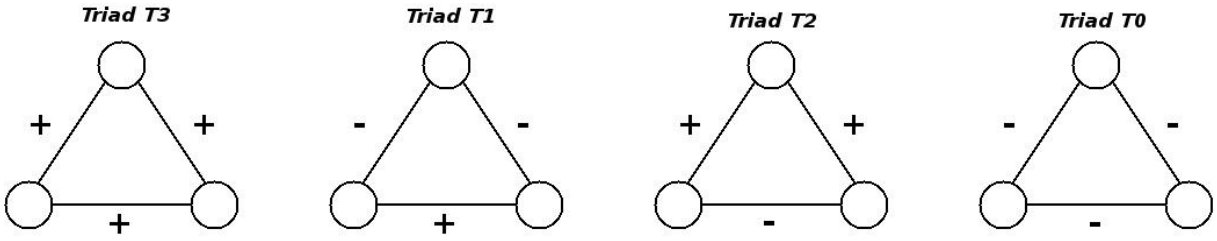

Figure 4.3: Triangle relationships supported by the structural balance theory.

trust the tuples that they generate accordingly. Thus, a pattern that is not selective will have a low weight. The tuples generated by such patterns will be discarded, unless they are supported by selective patterns. We also only keep tuples with high confidence. The confidence of the tuple is a function of the selectivity and the number of the patterns that generated it. Intuitively, the confidence of a tuple will be high if it is generated by several highly selective patterns.

As an initial filter, we eliminate all patterns supported by fewer than a specific threshold of seed tuples. We then update the confidence of each pattern, checking each candidate tuple $t =< e_1, e_2 >$ that is generated by the pattern in question.

For each candidate tuple, we check if there exists a set of high confidence previously extracted tuples for e_1 (e.g., $< e_1, e_x >$, $< e_1, e_y >$). If e_2 is equal to either e_x or e_y, then the tuple t is considered a correct match for the pattern, and an unknown match otherwise. Additionally, if the candidate tuple matches with a known negative example tuple (i.e., a tuple created by a set of seeds given initially as representing the oposite type of relation), the tuple t is considered an incorrect match for the pattern and the confidence of P is decreased. More formally, my algorithm adapted from Snowball defines $\mathrm{Conf}(P)$, i.e, the confidence of a pattern P, as follows:

$$\mathrm{Conf}(P) = \log_2(P_c) \times \frac{P_c}{P_c + P_u \times w_u + P_i \times w_i} \tag{4.6}$$

In the formula, P_c is the number of correct matches for P , P_u is the number of unknown matches, and P_i is the number of incorrect matches, adjusted respectively by the w_u and w_i weight parameters. The confidence scores are normalized so that they are between zero and one.

The approach that I used to calculate the pattern's confidence is also different from that used in the original Snowball. In the original approach, the authors only use the P_c and P_i, to calculate the pattern's confidence. More formally, the original Snowball defines $\mathrm{Conf}(P)$ as follows:

$$\mathrm{Conf}(P) = \log_2(P_c) \times \frac{P_c}{P_c + P_i} \tag{4.7}$$

In the original version of Snowball, for each candidate tuple, we check if there exists a set of high confidence previously extracted tuples for e_1 (e.g., $< e_1, e_x >$, $< e_1, e_y >$). If e_2 is equal to either e_x or e_y, then the tuple t is considered a correct match for the pattern, and an incorrect match otherwise.

We can also use another approach to calculate the pattern's confidence, using estimated relations. These relations can be predicted with basis on a social psychology theory, namely the structural balance theory [Heider, 1946], which has been proposed to reason about how different patterns of support and

opposition links provide evidence for the expression of different kinds of relationships. This theory has been shown to hold both theoretically and empirically for a variety of social community settings.

Structural balance considers the possible ways in which triangles on three individuals can be signed – see Figure 4.3. The theory posits that triangles with three positive signs (i.e., three mutual friends) and those with one positive sign (two friends with a common enemy) are more plausible, and hence should be more prevalent in real networks, than triangles with two positive signs (two enemies with a common friend) or none (three mutual enemies). Balanced triangles with three positive edges exemplify the principle that *the friend of my friend is my friend*, whereas those with one positive and two negative edges capture the notions that *the friend of my enemy is my enemy*, *the enemy of my friend is my enemy*, and *the enemy of my enemy is my friend*. The structural balance theory has also been developed extensively since the initial proposal, including the formulation of a variant named weak structural balance, proposed by Davis in the 1960s as a way of eliminating the assumption that *the enemy of my enemy is my friend* [Davis, 1967]. In particular, weak structural balance posits that only triangles with exactly two positive edges are implausible in real networks, and that all other kinds of triangles should be permissible.

Based on the idea that triangles with three positive signs and with one positive sign (two friends with a common enemy) are plausible, we form triangle relations composed by three entities where two of relations are know and follow the theory to predict the third relation. For example, if have three entities e_1, e_2, and e_3 where e_1 and e_2 have a support relations, and e_2 and e_3 have an opposition relation, then we will say that e_1 and e_3 have a opposition relation. Using these ideas to calculate the pattern's confidence, I can define $\mathrm{Conf}(P)$, as follows:

$$\mathrm{Conf}(P) = \log_2(P_c) \times \frac{P_c + E_c \times w_{ec}}{P_c + P_u \times w_u + P_i \times w_i + E_c \times w_{ec} + E_i \times w_{ei}} \tag{4.8}$$

In this version of the confidence formula, we also use E_c as the number of correctly estimated relationships for P, and E_i as the number of estimated relationships calculated using the opposite seeds given initially (i.e., I check if one of the three relations in the triangle is given in the opposite seed set, and check if one of the theories is verified), adjusted respectively by the w_{ec} and w_{ei} weight parameters.

The confidence of the extracted tuples is calculated as a function of the confidence values for the patterns and the number of patterns that generated the tuples. Intuitively, $\mathrm{Conf}(t)$, i.e., the confidence of an extracted tuple t, will be high if t is generated by several highly selective patterns. More formally, the confidence of t is defined as follows:

$$\mathrm{Conf}(t) = 1 - \prod_{i=0}^{|P|} (1 - (\mathrm{Conf}(P_i) \times \mathrm{Match}(C_i, P_i))) \tag{4.9}$$

In the formula, P is the set of extraction patterns that generated t, and C_i is the context associated with an occurrence of s that matched a specific pattern P_i with degree of match $\mathrm{Match}(C_i, P_i)$. The $\mathrm{Match}(C_i, P_i)$ value represents the biggest similarity value obtained when we compare the tuple with each one of the pattern's centroids. After determining the confidence of the candidate tuples, the algo-

rithm discards all tuples with low confidence (i.e., those with a score below a given threshold), because these tuples can add noise into the pattern generation process, which would in turn introduce invalid tuples.

The Snowball system is thus defined by a set of parameters, that are used to regulate the system's functioning. With these parameters we can regulate similarity values, weight values, or even define if we want extract directed relations or not. The parameters are as follows:

- **occ_both_directions:** Find occurrences in both directions. This parameter is used to enable or disable the extraction of bi-directional relations.
- **max_tokens_away:** This value represents the maximum number of tokens between two entities in a specific sentence, so that this sentence can be used in the relation extraction process, for that specific pair of entities.
- **min_tokens_away:** Contrary to the above parameter, this parameter refers to the minimun number of tokens between two entities in a specific sentence.
- **context_window_size:** Number of tokens used before the first entity and after the second entity, to use in the relation extraction process. The value of this parameter takes into account that the stop words have been already removed.
- **number_iterations:** Maximum number of iterations of the system. By default the system stops if an iteration does not add any new tuples. If each iteration extracts new tuples, the system stops when the iteration number is the same as this parameter.
- **min_degree_match:** This value represents the minimum threshold similarity between a possible tuple that we want extract and the centroid of each cluster (i.e., pattern). If the similarity between a tuple and a cluster is above this threshold, then the tuple is considered as a tuple extracted by that pattern.
- **min_tuple_confidence:** The confidence of a candidate tuple, calculated with the patterns that generated the tuple, represents the minimum value so that the tuple can be used as a seed for the next iteration. If a tuple has enough confidence, then its entities are considered as seeds for the next iteration.
- **min_pattern_support:** Minimum number of patterns that generated a tuple so that the tuple can be used in the clustering phase. If a tuple is extracted by a number of patterns below this value, then the tuple is discarded from the other phases of the process.
- **DBScan_min_points:** Minimum number of tuples required to form a cluster (i.e., a pattern). In the clustering phase, if a cluster has less that this number of tuples, then the cluster is discarded and not considered for the tuple generation stage.
- **DBScan_eps:** Minimimum similarity value between a tuple and other tuples to form a cluster. This value is used in the clustering phase and represents the threshold similarity between a tuple t_1

generated by the set of seeds and the other tuples generated by the same set of seeds. If the similarity is above this threshold, then we consider that these tuples are neighbours of t_1.

- **weight_left_context, weight_middle_context, weight_right_context:** Weight of the feature vector built to represent the context before the first entity, between the two entities, and after the second entity, respectively. These values represent the importance of each one of these context windows in the similarity calculation.

- **w_i, w_u:** Weight parameters for the number of unknown matches and the number of negative matches of each pattern. These weights are used to manage the importance of the unknown and negative matches in the calculation of a pattern's confidence.

- **w_ei, w_ec:** Weight parameters for the number of correct and incorrect estimated relations based on the structural balance theory. These weights are used to manage the importance of these parameters in the calculation of the confidence of patterns.

4.4 Building Signed Networks

We can extract multiple relations between the same pair of entities. In this case, we aggregate all the relations corresponding to the same entity pair, assuming that the classification for the relation is the majority class in all the relations (e.g., if the same pair of entities has three support relations and one opposition relation, we assume that they have a support relation).

A graph is finally built from the candidates, by creating an arc for each aggregate from the previous step, giving it a sign according to the classification that was made for the aggregate. Optionally, the graph can be filtered so as to only consider edges whose support (i.e., the number of candidates in the aggregate) is above a given threshold, or whose semantic orientation score is above a given confidence.

4.5 Experimental Results

The network extraction method that was outlined in the previous section was applied to newswire corpora in the Portuguese language, namely to news articles published on the online version of *Público*, from the last 5 years. The Portuguese corpus consists of a total of 176.865 Portuguese news stories containing 2.452.713 sentences, maintaining a total of 244.760 different persons (after disambiguation). Figure 4.4 shows the regular and cumulative distributions for the total of news articles published in each year, as well as for the number of sentences analyzed by Snowball. The same figure also shows the distributions of the total number of person names that are mentioned in those years, after the disambiguation step.

When assessing the quality and veracity of the results for the support and opposition relations that were extracted, I conducted an empirical analysis and relied on profile information, due to the fact that there are not strict parameters or ground-truth lists to truly assess the relation between persons. In order

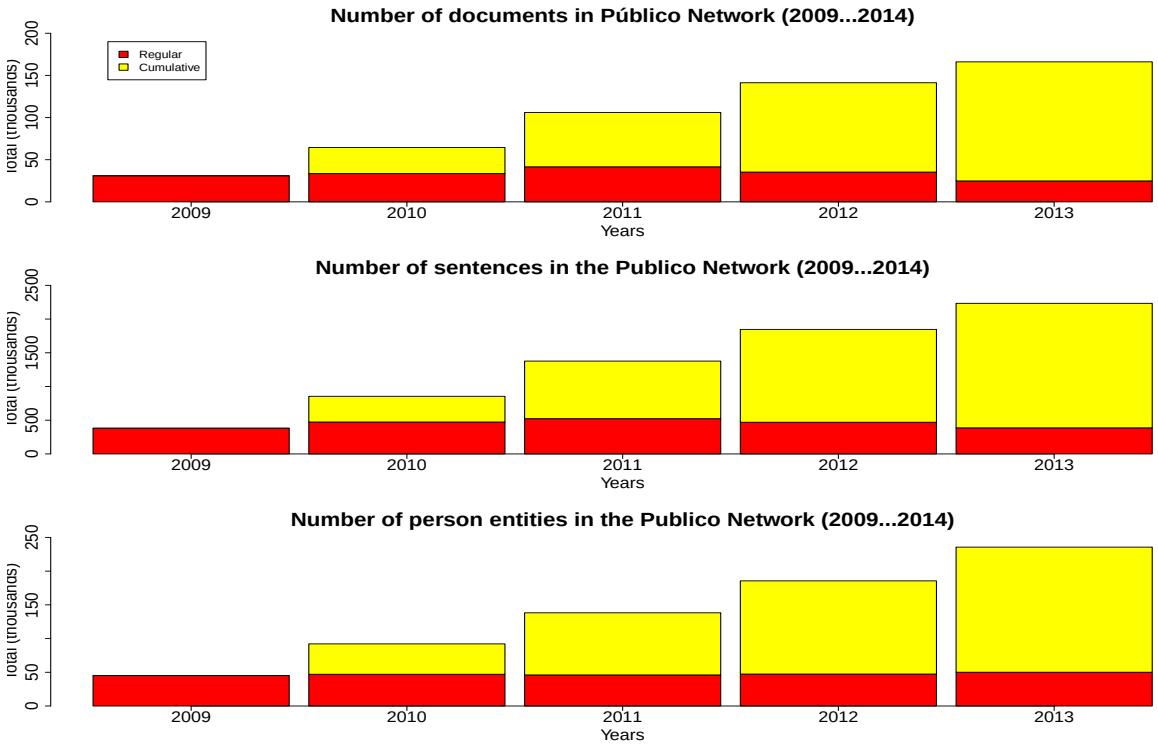

Figure 4.4: Number of news articles, sentences, and of person entities in the Publico dataset.

to evaluate the relations that were extracted, I also built two different ground-truth datasets automatically, based on a list with all the Portuguese politicians which seat in the parliament and that appear in the articles published on the *Público* newspaper, and with the respective parties to which they belong, assuming that:

1. All the political entities that belong to parties with the same orientation (i.e., left or right political leans) have a support relation between them, and they are assumed to have an opposition relation towards the persons with the other orientation. This dataset is composed by 5264 support relations and 20 opposition relations.

2. All the political entities that belong to the same party have a support relation between them, and an opposition relation towards the persons on the other parties. This dataset contains more relations, with 4401 support relations and 618 opposition relations.

I used the proposed procedure to extract support and opposite relations, performing an extensive set of experiments where I tested the system 350 times. Each one of these tests represents a different combination of the parameters explained in Section 4.3, where all have fixed values except four parameters, i.e, min_tuple_confidence, min_degree_match, min_pattern_support, and DBScan_eps. In sum, the system was tested with all the possible combinations of the parameters presented in Table 4.1.

The extracted relations are evaluated through some of the common metrics that have been used in previous works in the area of relation extraction from text. Specifically, I used precision, recall, the F_1 measure, and accuracy.

parameter	value	parameter	value
occ_both_directions	1	min_tuple_confidence	0.2 - 0.8
max_tokens_away	8	min_pattern_support	1 - 2
min_tokens_away	2	DBScan_eps	0.2 - 0.6
context_window_size	6	weight_left_contex	0.2
number_iterations	8	weight_middle_context	0.6
min_degree_match	0.2 - 0.6	weight_right_contex	0.2

Table 4.1: Paramater values used in the relation extraction tests.

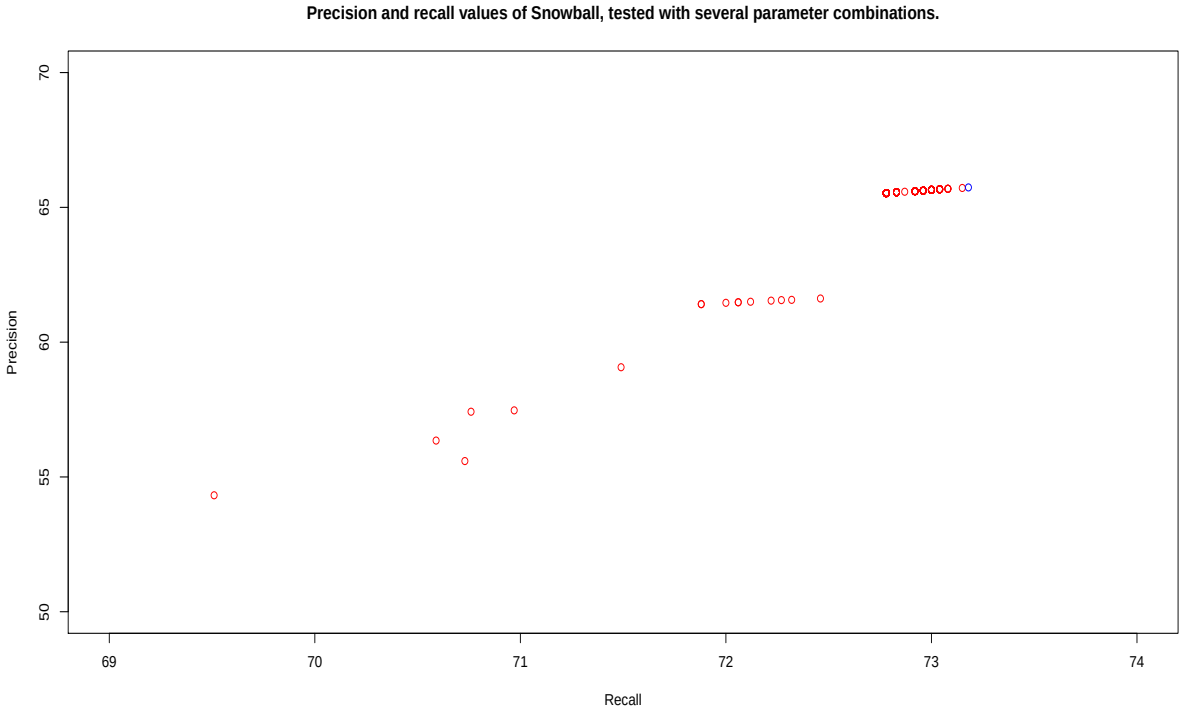

Figure 4.5: Results obtained by Snowball with various combinations of the parameters when testing with the ground truth dataset based on left/right ideologies.

Figures 4.5 and 4.6 present the obtained results for the two ground truth datasets, where each point represents the obtained values in one of the experiments, in terms of the precision and recall measures. The bold points refer that more than one experiment have returned the same values. The blue point represents the result with higher F_1. The results presented in Figure 4.6 do not represent the quality of the system. The reason of these results is the small number of opposition relations in the dataset about the left/right ideologies because when evaluating the opposition relations, most of the experience have 1 correct relations extracted (i.e., they do not extracted almost any of the 20 opposition relations) and when the average of the precision is computed, the value of the opposition relations is low and this in fact decrease the overall precision.

Results showed that best results in both datasets have the parameters 0.6 in min_tuple_confidence, 1 in min_pattern_support, 0.6 in min_degree_match, and DBScan_eps with a value of 0.5, for both experiments, returning a F_1 score of 69.26% and 77.83% respectively. Observing the result, one can conclude that results with higher precision have also higher recall.

Apart from the results obtained with these dataset, I observed some of the results manualy in order

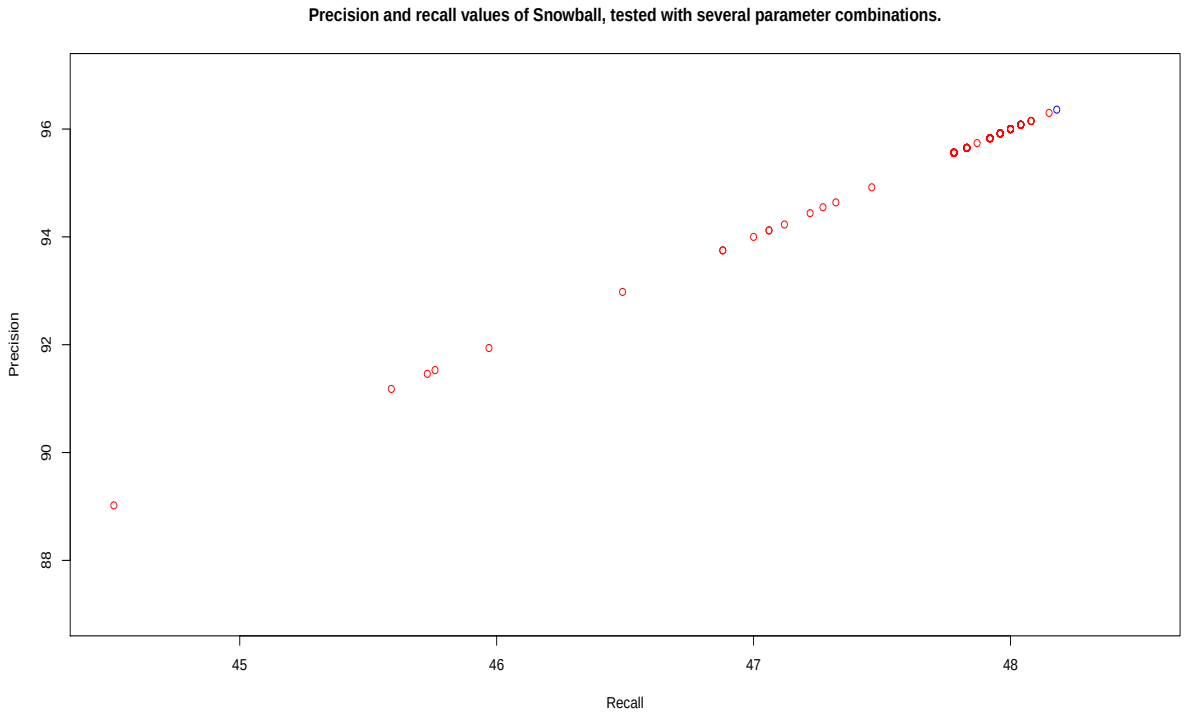

Figure 4.6: Results obtained by Snowball with various combinations of the parameters when testing with the ground truth dataset based on party affiliations.

	My Snowball	Original Snowball
Number of nodes	588	721
Number of oppositive arcs	376	464
Number of support arcs	1248	1395
Number of arcs	1624	1859
Min Tuple Confidence	0.6	
Min Pattern Support	1	
Min Degree Match	0.6	
DBScan Eps	0.5	

Table 4.2: Nodes and arcs of the resulting network and the parameters that generated it.

to analyse the relations extracted that are not contained in the datasets. The min_tuple_confidence is the most relevant parameter, affecting considerably the number of relations extracted and the precision. Experiments with high min_tuple_confidence values tend to extract few relations but in other hand have higher precision, while experiments with low values of min_tuple_confidence extract several relations but the precision is below the others, probably because in cases where the parameter's confidence is low, the system use more sentences, many of them noisy that will influence the negatively the pattern's confidence.

I then used the proposed procedure, with the best performing parameters, to build a signed network from the Portuguese news dataset, afterwards performing a statistical characterization for the the network that was generated. Table 4.2 presents the number of nodes and relations extracted in the experiment with better results, as well as the parameters used to obtain these results.

Table 4.3 presents the best evaluation results obtained with both datasets, comparing them with the

		Dataset 1	Dataset 2
Original Snowball	Precision	50%	64.49%
	Recall	43.67%	65.87%
	F_1 Measure	46.62%	65.04%
	Accuracy	96.43%	96.17%
My version of Snowball	Precision	65.74%	82.35%
	Recall	73.18%	73.78%
	F_1 Measure	69.26%	77.83%
	Accuracy	96.71%	96.16%

Table 4.3: Evaluation results over the two ground truth artificial datasets.

results obtained with the original version of Snowball, when using the same values for the parameters. Observing the results, one can conclude that the adaptations made on the Snowball system, improved the results when extracting support and opposition relations.

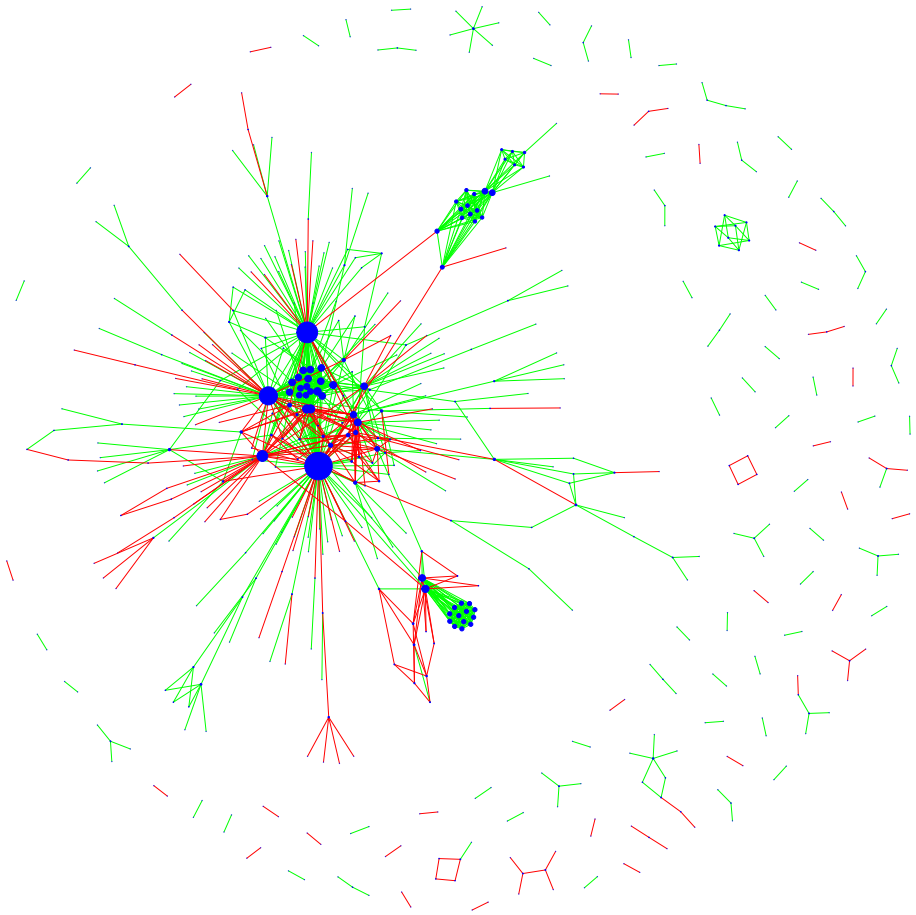

Figure 4.7: Signed Network with the support and opposition relations extracted from Público.

Finally, Figure 4.7 shows the signed network built, where the blue nodes represent to names of persons and the edge represent the a polarity relation between two persons. The size of each node is proportional to the number of relations of each entity, i.e., if a specific entity have several relations with other entities, then its size will be bigger than a entity that have one or few relations. The color of the

Figure 4.8: Signed Network built based on Público, having the nodes more than 20 relations.

edge describes the sign of the relation, the green represent a support relation and the red represent a opposition relation. Figure 4.8 shows a sample of the network corresponding to the entities with more than 20 relations extracted, presenting the disambiguated name of each entity.

4.6 Conclusions and Critical Discussion

In this chapter, I have shown that natural language processing techniques can be used effectively to extract signed social networks from newswire documents. I specifically proposed a method that derives networks from meaningful co-occurrences of person names within individual sentences, and that classifies the co-occurrences according to their semantic polarity orientation.

I reported on a large-scale extraction from newswire documents related to politics, written in Portuguese. In order to evaluate the proposed method, I used two ground-truth datasets, based on a list with all the Portuguese politicians and their political orientations. It was interesting to see that the adaptations made on the Snowball system, improved the results when extracting support and opposition relations. Although the results do not represent a evaluation for all the relations extracted because some

of the relations are not in the evaluation datasets. It would be interesting if I had a dataset with all the support and opposition relations in the 5 years of *Público* analyzed instead of create two datasets based on theories. For future work, it would be interesting to conduct a detailed analysis of the resulting networks focusing for instance on the identification of network clusters or influential nodes.

Chapter 5

Extracting Part-of Relations Between Location References

This chapter presents a method for extracting *part of* relations between locations mentioned in books. I used basically the same bootstrapping method explained in Chapter 4, introducing some differences related to the fact that in this case we want extract directed relations. I also report on extractions from English fiction books.

The following section presents the problem and an overview of the relation extraction system. Section 5.2 presents the relation extraction system in detail, and the differences relatively to the approach explained in the previous chapter. Section 5.3 presents the experimental results. Finally, Section 5.4, presents the main conclusions, together with a critical discussion.

5.1 Introduction

The content of books often is pervaded by information of a geographical or spatial nature, particularly location information such as addresses, postal codes, and so forth. It is natural to assume that associations exist between a specific location and other locations. Geographical books, for instance, can be very rich in location information, allowing for example the use of text mining techniques to extract relations between locations. Analyzing and interpreting geographic information involves seeking meaningful patterns, relationships, connections, and processes.

The nascent research area of literary geography/literary cartography, which aims at visibly rendering complex overlays of real and fictional geographies, can also stand to benefit from computational information extraction approaches. Literary geography includes several overlapping perspectives following the main epistemological and theoretical turns in the fields of human and cultural geographies. With the rising interest in regionalism, literary geography is not just addressing geographic analysis of literature, but rather giving a helping hand in descriptive geographic portrayals. Regionalist, humanist, and socially critical perspectives diversified the ways literature could be used in analytic terms and thereby turned literature into an object of study. The field of geographic studies of literature has been categorized the-

matically by following the development of research in human and cultural geographies in a somewhat chronological manner.

The geography of fiction also follows its own distinct rules, since literature can create any space, without physical restrictions. The distinctive tools of literary writing include the ability to destabilize taken-for-granted geographies. It belongs to the ambitious goals of literary geography to find out more about those rules and to demonstrate that the spatial dimension of fictional accounts can actually be one key to the understanding of the whole plot behind particular texts.

In other hand, while literary geography is an overall broad, literary cartography provides one possible approach by using a symbolic language. Spatial elements of fictional texts can be translated into cartographic symbols, which allows new ways in exploring and analysing the particular geography of literature. Literary cartography can be divided into two main branches, that are closely linked: the mapping of a single text and its spatial elements, and the mapping of groups of texts, leading ultimately to statistical and quantitative approaches.

In this work, I present a method for extracting *part-of* relations from fiction books. The relations are extracted from meaningful co-occurrences of location names within individual sentences (i.e., co-occurrences that are associated to specific linguistic patterns, commonly used to encode part-of relations). In order to perform the extraction of relations between locations, I again used a bootstrapping method, through an adapted version of the previously proposed Snowball approach. As explained before, the Snowball system builds patterns in order to extract relations in one dataset, according to the seeds given initially. In this case, the system is first used to generate patterns and extract *part-of* relations in geographical books, and then it is used to extract the same kind of relations between locations, using the patterns that were originally discovered also on the fiction books.

To test the quality of the system, I report on a case study involving a geographical book describing the United Kingdom [Gardiner, 1999] to generate patterns and extract relations, I also report on experiments with two series of fiction books, to extract relations using the previously generated patterns. These series correspond to the Bas-lag trilogy of Mieville [2002, 2003, 2004] and to the Gormenghast trilogy by Peake [2011]. To evaluate the proposed method, we evaluate the United Kingdom relations that were extracted based on a geographical web database called GeoPlanet[1], that contains all the part-of relations regarding places in the *United Kingdom*.

5.2 Adapting Snowball for Extracting Part-Of Relations

Initially, we split the books into sentences and apply the NER model, that is already distributed with Stanford NER, in order to recognise the location entities presented in these sentences. These sentences will be used to extract *part of* relation between the locations present in them.

Then I used basically the same bootstrapping method explained in Chapter 4, with a few changes. The main difference is related to the use of direction in the relations, being all the other differences caused by this one. The seeds given initially have direction, i.e a seed $< e_1, e_2 >$ is completely different

[1] `https://developer.yahoo.com/geo/geoplanet/`

from a seed $< e_2, e_1 >$. For instance, having a relation London *part of* United Kingdom is different of having United Kingdom *part of* London. Given this, all the seed and extracted tuples need to have the same direction, if e_1 *part of* e_2 then all the other seeds need a direction given by the first entity towards the second.

Before each iteration we compute the transitive closure of the seeds, in order to find new seeds (e.g., if we have extracted two seeds <United Kingdom, England> and <England, London>, then we will add a need seed to the set of extracted seeds, more precisely <United Kingdom, London>). With basis on this, in the next iterations we will probably generate more patterns and extract more seeds.

Finally, the most important change relatively to the system presented before is the formula for the pattern's confidence calculation. In this case we use the first confidence formula presented in Section 4.3 (i.e., Formula 4.6, where we check for each candidate tuple $t =< e_1, e_2 >$ if these exists a set of high confidence previously extracted tuples for e_1 (e.g., $< e_1, e_x >$, $< e_1, e_y >$). If e_2 is equal to either e_x or e_y, then the tuple t is considered a correct match for the pattern, and an unknown match otherwise.

Although the correct and unknown matches are the same as in the previous system, the incorrect matches are different. If the candidate tuple matches with a known tuple s that was previously extracted, where $s =< e_2, e_1 >$, then the tuple t is considered an incorrect match for the pattern. We do not need a set of seeds representing an opposite relation, because when we are extracting directed relations the incorrect matches occurs when we extract tuples have an opposite direction. More formally, we define $\mathrm{Conf}(P)$, i.e, the confidence of a pattern P, as follows:

$$\mathrm{Conf}(P) = \log_2(P_c) \times \frac{P_c}{P_c + P_u \times w_u + P_i \times w_i} \tag{5.1}$$

This formula is similar to one of the confidence formulas presented before, where P_c is the number of correct matches for P , P_u is the number of unknown matches, and P_i is the number of incorrect matches, adjusted respectively by the w_u and w_i weight parameters.

5.3 Experimental Results

This subsection describes the experimental validation performed on the part-of relation extraction process, namely a complete set of experiments that evaluated the system's performance on extracting relations between location references for the English language.

In order to evaluate the system, I used a geographical book regarding the *United Kingdom* [Gardiner, 1999]. This book consists of a total of 6439 sentences with a total of 485 different locations.

To validate the geographic places and identify the relations between some of them (i.e., in this case, we are interested exclusively in the part-of relations), I used a specific tool, namely the Yahoo! GeoPlanet. In practical terms, GeoPlanet is a resource for managing all named places on Earth. By using GeoPlanet, we can traverse the global spatial hierarchy of administrative places. In this case, we use the tool to explore the place hierarchy, for instance for a location *London*, we want the location's parents, i.e. *England*, *United Kingdom* and *Europe*.

First, I extract all the different locations present in the *United Kingdom* geographical book used to extract the relations, and then used the Yahoo! GeoPlanet tool to search for all the *belong-to* relations for each different location. In the resultant relations, I applied the transitive closure to obtain some relations that possibly were not specified in Yahoo! GeoPlanet.

This dataset contains all the part-of relations for the places mentioned in the book, but some of them are not expressed in the book. For instance, if *London* is a place present in the book, with the Yahoo! GeoPlanet tool and the transitive closure technique, I will create a relation in the dataset between *London* and all the possible parents (e.g., *Europe*), but if any sentence in the book contains both locations (i.e., *London* and *England*), then the dataset contains an entity pair that is not expressed in the book. To resolve this, I extract all the location entity pairs $< l_1, l_2 >$ present in the book, where the l_1 and l_2 appear in the same sentence. Finally, I crossed the location entity pairs of the book with the entity pairs of the dataset, obtaining a dataset with all the part-of relations present in the book.

The relations extracted through the proposed approach are then evaluated through the same common metrics that were used in the evaluation of the relation extraction system explained in the previous chapter. Specifically, precision, recall, and the F_1 measure.

I used the proposed procedure to extract *part-of* relations from the English book dataset, in this case also performing an extensive set of experiments where I tested the system 350 times, with the same parameter combinations explained in the previous chapter.

Figure 5.1 shows the obtained results, where each point represents the obtained values in one experiment, in terms of precision and recall measures. The blue points represent the 10 results with the higher F_1 scores, with the parameters oscillating between a 0.3 or 0.4 in min_tuple_confidence, a min_pattern_support in 1 and the other two parameters with a different value in at least one of the 10 experiments. The best results obtained, being the best result a combination of all the fixed parameters together with a 0.2 value in the min_tuple_confidence, 1 in min_pattern_support, 0.6 in min_degree_match, and DBScan_eps with a value of 0.4, returning a F_1 of 19.2%.

Analysing the results, we also conclude that the parameter with most influence in the results is the `min_tuple_confidence`. Figure 5.2 shows the influence of this parameter in the results, having all the other parameter with a fixed value, according to the result with higher F_1 in the previous experiment.

As for the second set of experiments, the main objective is to show that the patterns that extract the relations between the UK places, can be used to extract part-of relations in another context (e.g., fiction books). More precisely, I used a collection of fiction books, composed by the Bas-lag trilogy by Mieville [2002, 2003, 2004], namely *The Scar*, *Perdido Street Station* and *Iron Council*, and the Gormenghast trilogy by Peake [2011], namely *Titus Groan*, *Gormenghast* and *Titus Alone*, to apply the patterns generated before and extract the *part-of* relations between fictional places, consisting in a total of 14.870 sentences, and a total of 106 different locations.

In order to do the second set of experiments, I ran the relation extraction system two times. First, I ran the system giving the UK geographical book and a set of directed seeds as input for generating the *part-of* relational patterns. Second, I ran the system again, giving as input one of the considered fiction books and the set of patterns generated before. In this second time, the system does a single iteration,

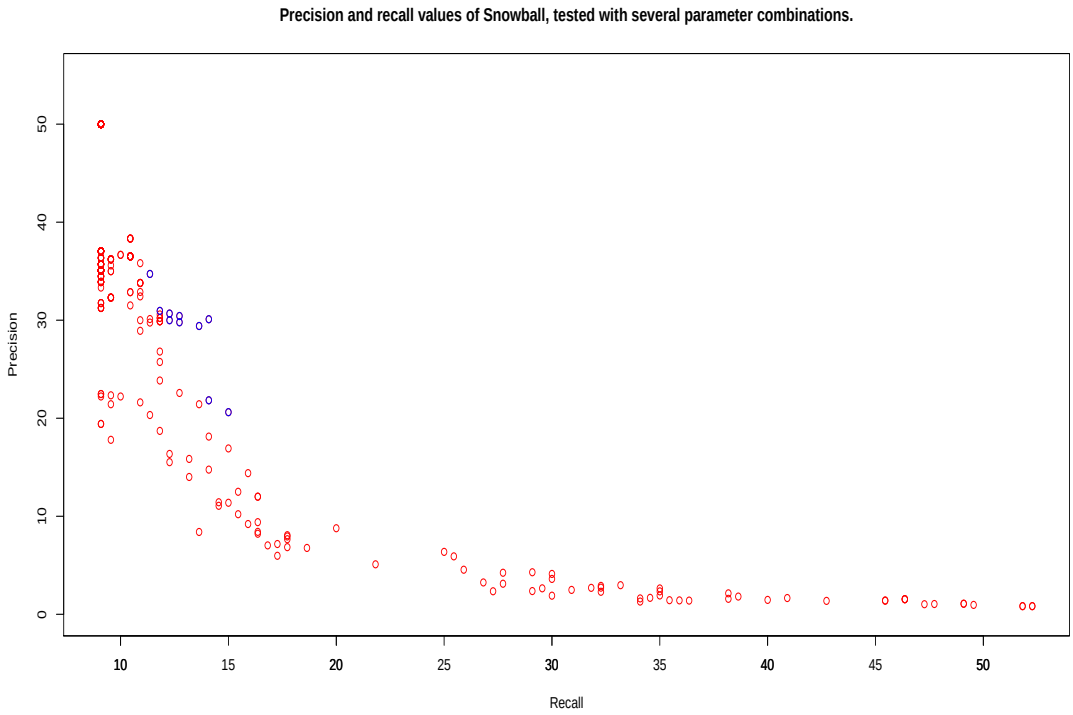

Figure 5.1: Results obtained by Snowball with various combinations of the parameters.

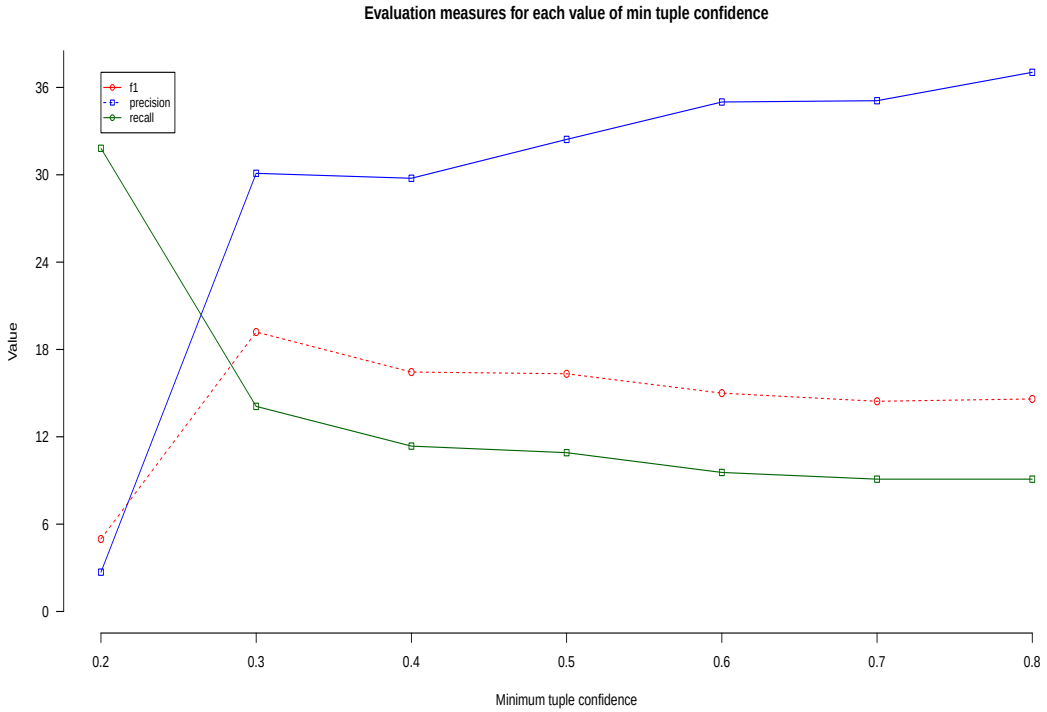

Figure 5.2: Results obtained by varying the minimum tuple confidence parameter.

because we cannot continue to calculate the confidence of patterns without the tuples that generated it. In this iteration, we jump the first steps of finding occurrences of seed tuples and generate patterns, because the occurrences of seed tuples are used to generate patterns, and in this case patterns are

given initially and we do not generate any.

Book series	Sentences	Locations	Unique location pairs	Extracted relations
Mervyn Peake, The Illustrated Gormenghast Trilogy	24330	56	12	2
China Mieville, Bas-lag Trilogy	49932	266	127	27

Table 5.1: Tuples extracted from the fiction book.

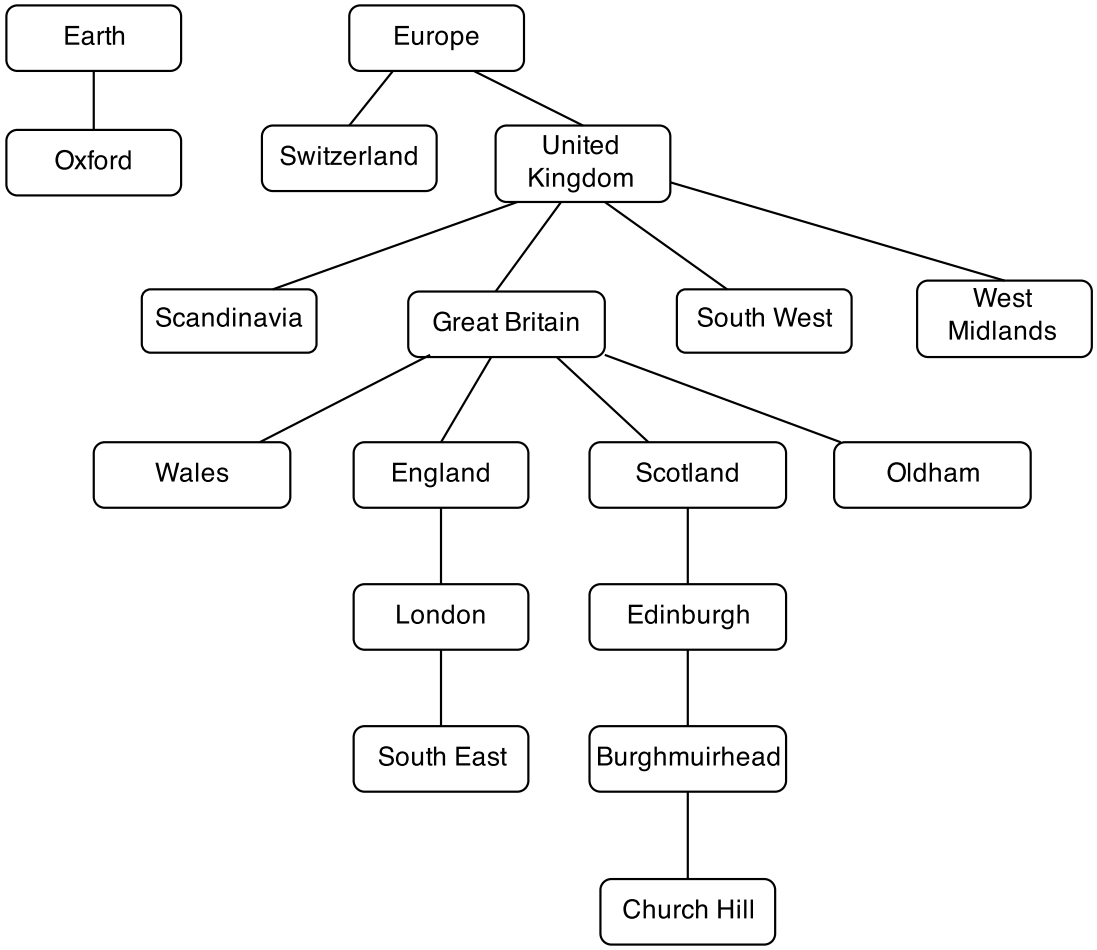

Figure 5.3: Hierarchy of the UK places extracted from the book.

Table 5.1 shows a characterization for the relations extracted by the system, where we can seed that few relationships where in fact extracted. Despite this short value of relations extracted, the *unique location pairs* column in the table also contains a short value. This column present all the pairs present in the books, even pairs with no relations (i.e., *London*, *Manchester*, etc.) that are very common in this kinds of books. With this, we can conclude that there were not many relationships to extract, and the number of fiction relations extracted are not bad at all.

Figures 5.3 and 5.4, present a graphical representation of the UK relations extracted from the UK geography book and the hierarchy of places in the two trilogies of fiction books, respectively, where the bottom places are part of the top ones. Looking at both trees, we can conclude that the relations

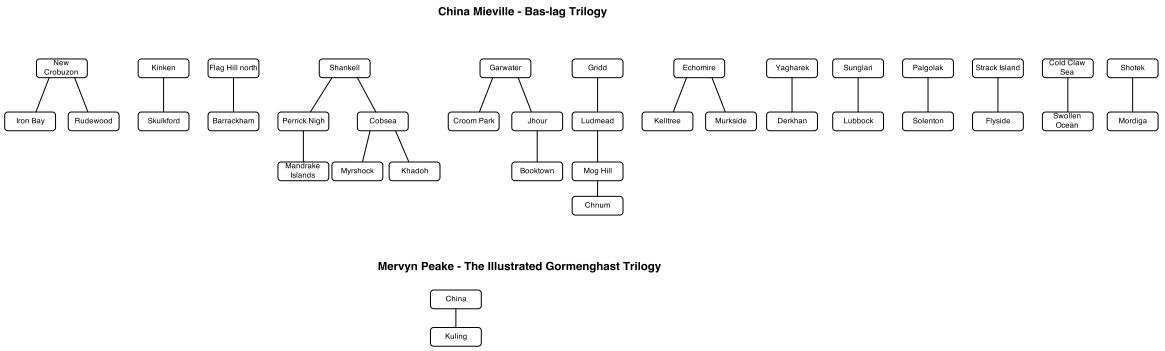

Figure 5.4: Hierarchy of the fiction places extracted from the two trilogies of fiction books.

extracted from the UK geography book have better results comparing to the fiction relations extracted. Some of the relations extracted from the fiction books were found to be incorrect after a manual inspection (e.g., <Cobsea *part-of* Khadoh>, <Strack Island *part-of* Flyside>).

5.4 Conclusions and Critical Discussion

In this chapter, I have shown that natural language processing techniques can be used effectively to extract relations between location entities mentioned in books. I proposed a method that extracts relations from meaningful co-occurrences of location names within individual sentences, classifying these relations according to part-of relations.

I reported on extractions from books related to geography and fiction, written in English. In order to evaluate the proposed method, I used a dataset based on a geographical web database called GeoPlanet[2]. This dataset was used to evaluate the relations extracted from the geography book, and it was interesting to see that the best results are obtained when we have a low tuple confidence but a higher pattern confidence.

I also evaluated the system on collections of fiction books, using the best parameters from the first experiment. In this experiment, I used the generated patterns of the previous experiment to extract tuples in the fiction books. Despite the short number of relations that are extracted, the pairs present in the books that maybe contain a possible relations are not many. With this, we can conclude that there were not many relationships to extract, and thus the results are of an acceptable quality.

[2] `https://developer.yahoo.com/geo/geoplanet/`

Chapter 6

Conclusions and Future Work

This dissertation presented the research work that was conducted in the context of my MSc thesis. I described a relation extraction approach capable of extracting support and opposition relations between persons in news documents written in Portuguese, and capable of extracting relations between locations in books written in the English language.

My study on the relation extraction task provided some interesting contributions, as we have that few previous works have evaluated relation extraction performance in languages other than English, leaving several open questions for those trying to develop such systems. In my MSc thesis, I have presented and thoroughly evaluated a bootstrapping approach for relation extraction, which uses an adapted version of the well-know Snowball system.

6.1 Main Contributions

Through a series of experiments, my work provided the following main contributions:

1. In order to address the fundamental baseline task of Named Entity Recognition (NER), I created a model for the Portuguese language. With this model, the system became able to recognize entity mentions in textual documents written in Portuguese, with an considerably high accuracy (i.e., an average F_1 score of 68.06%).

2. With the objective of trying to improve the efficiency of the recognition of persons which can be mentioned with different names, I developed an heuristic method that automatically disambiguates entity persons, clustering several names which reference the same real world entity. The heuristics are based on dividing the persons by gender, using male and female list names, and on the removal of labels present in the entity name. I also only disambiguate names with more than one appearance in the texts. The remaining names were compared with all the other names with the same gender, using the simple similarity procedure based on the Jaro-Winkler TF-IDF similarity measure.

3. I developed a relation extraction approach that introduces several adaptations over *Snowball*. This

system uses a document collection and a set of seeds to extract relations between person entities, and relations between location entities. I improved the confidence formulas of the original Snowball, used a different clustering methodology to cluster the tuples and applied a social psychology theory, more precisely the structural balance theory, to evaluate the existing relations. The structural balance theory considers the possible ways in which triangles on three individuals can be signed. The approach allows one to find new relations, as well as evaluate if the real relations that we extract are correct or incorrect. This technique is based on the principles *the friend of my friend is my friend*, and *the enemy of my enemy is my friend*.

4. I performed an extensive set of tests to evaluate the relation extraction system when extracting relations between persons. In order to perform this evaluation, I built two different ground-truth datasets automatically, based on a list with all the Portuguese politicians which seat in the parliament, and the respective parties to which they belong, assuming that:

   - Political entities that belong to parties with the same orientation (i.e., left or right) have a support relation between them, and we can assume that they have opposition relations towards the persons with the other orientation.
   - Political entities that belong to the same party have a support relation between them, and an opposition relation towards the persons on the other parties.

   Results with these experiments revealed an average F_1 score of 69.26% for the dataset based on left/right ideologies and average of 77.83% for the dataset based on party affiliations.

5. I also developed a dataset containing part of relations between locations, in order to evaluate the performance of the relation extraction system when extracting relations between locations, by using the Yahoo!GeoPlanet[1] service. The dataset contains all the *part of* relations where the two entities appear in a geographic book Gardiner [1999]. Results with these experiments revealed an acceptable quality with an average F_1 score of 19.02%.

6.2 Future Work

Despite the interesting results, that are also many open challenges for future work. It would be interesting, for instance, to experiment with the usage of the sorted neighbourhood method [Hernández and Stolfo, 1998] in the named entity disambiguation step. The use of this method will reduce the comparisons between candidate names that will lead to a performance improvement. A possible solution to this approach is through a set of keys instead of the normal procedure where only one key is used. The use of only one key will probably not give good results, for instance, if we have Jose Socrates and Socrates, and a key composed by the first three characters of each name, the keys will be `JOSSOC` and `SOC`, respectively. Having a big amount of names to disambiguate, even with a big window value these entities will probably never be compared, however both represent the same person.

[1] `https://developer.yahoo.com/geo/geoplanet/`

The experiments reported in this dissertation have mostly addressed relation extraction between named entity references, assuming the existence of a named entity recognition system. For future work, I believe that the usage of an anaphora resolution system would be interesting, in order to discover sentences in the collection of documents that are currently not being considered because they did not refer directly to any person name. For example, in the sentence ***He*** *criticized* ***him*** *for the way he talked*, we have an opposition relation between two persons, but this sentence is not considered by my system. With the application of a system capable of identifying the person regarding **He** and the person regarding **him**, we would have more sentences to analyse in the relation extraction process, and probably more relationships would be extracted.

It would be also interesting to experiment with the usage of additional features in the vector space model representation used in the relation extraction system. At this time, each sentence used in the relation extraction process has a feature vector representing its context, where the features are the words and the word clusters corresponding to each word. I believe that features derived from the morphological class of each word could provide particularly rich information for relation extraction purposes. These morphological classes could be recognized by a Parts-of-Speech (POS) tagger.

The signed networks that I extract through my method can now also support a large set of different analysis operations. One important analysis task is that of sign inference, i.e., the task of inferring unknown (or future) trust or distrust relationships between individuals, given a partially observed signed network [Leskovec et al., 2010b]. Previous works have proposed approaches that consider the notion of structural balance in signed networks, building inference algorithms based on information about links, triads, and cycles in the network. Another important task concerns with community detection in the networks that are automatically generated [Chen and Ji, 2010], particularly considering community detection methods for signed networks [Yang et al., 2007]. Currently ongoing work is specifically addressing the task of studying community evolution with basis on time-varying networks extracted from textual documents, using a methodology similar to that which was proposed by Palla et al. [2007].

Bibliography

E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, 2000.

F. Barreto, A. Branco, E. Ferreira, A. Mendes, M. Fern, A. B. D. Nascimento, F. Nunes, and J. R. Silva. Open Resources and Tools for the Shallow Processing of Portuguese, 2006.

D. Batista, D. Forte, R. Silva, B. Martins, and M. Silva. Extracção de Relações Semânticas de Textos em Português Explorando a DBpédia e a Wikipédia. *Linguamática - Revista para o Processamento Automático das Línguas Ibéricas*, 5(1), 2013.

M. Bautin, C. B. Ward, A. Patil, and S. S. Skiena. Access: news and blog analysis for the social sciences. In *Proceedings of the international conference on World Wide Web*, 2010.

E. Bick. *The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. University of Arhus, 2000.

P. Bramsen, M. Escobar-Molano, A. Patel, and R. Alonso. Extracting social power relationships from natural language. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2011.

A. Z. Broder. On the Resemblance and Containment of Documents. In *In Compression and Complexity of Sequences (SEQUENCES'97)*, 1997.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4), 1992.

M. Bruckschen, J. G. C. de Souza, R. Vieira, and S. Rigo. *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM, capítulo Sistema SeRELeP para o reconhecimento de relações entre entidades mencionadas*. Linguateca, 2008.

M. J. Brzozowski, T. Hogg, and G. Szabo. Friends and foes: ideological social networking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008.

N. Cardoso. REMBRANDT - Reconhecimento de Entidades Mencionadas Baseado em Relaes e ANálise Detalhada do Texto. In *Encontro do Segundo HAREM, PROPOR 2008*, 2008.

Z. Chen and H. Ji. Graph-based clustering for computational linguistics: a survey. In *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing*, 2010.

J. A. Davis. Clustering and structural balance in graphs. *Human Relations*, 20(2), 1967.

D. S. Day, J. S. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. B. Vilain. Mixed-initiative development of language processing systems. In *ANLP*, 1997.

G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel. The automatic content extraction (ace) program - tasks, data, and evaluation. In *LREC*, 2004.

P. Doreian and A. Mrvar. Partitioning signed social networks. *Social Networks*, 31(1), 2009.

D. K. Elson, N. Dames, and K. R. McKeown. Extracting social networks from literary fiction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2010.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.

O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. *Communications of the ACM*, 51(12), 2008.

L. C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1), 1977.

C. Freitas, D. Santos, H. G. Oliveira, P. Carvalho, and C. Mota. *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM.* Linguateca, 2008.

P. Gamallo, M. Garcia, and S. Fernández-Lanza. Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, 2012.

M. García and P. Gamallo. Evaluating Various Linguistic Features on Semantic Relation Extraction. In *Proocedings of the Conference on Recent Advances in Natural Language Processing*, 2011.

V. Gardiner. *Changing Geography of the UK: Third Edition*. Taylor & Francis, 1999.

A. Hassan and D. Radev. Identifying text polarity using random walks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2010.

A. Hassan, A. Abu-Jbara, and D. Radev. Extracting signed social networks from text. In *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing*, 2012.

V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 1997.

F. Heider. Attitudes and cognitive organization. *Journal of Psychology*, 21, 1946.

M. A. Hernández and S. J. Stolfo. Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. *data mining and knowledge discovery*, 2, 1998.

M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, 2004.

J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2004.

E. Key, L. Huddy, M. Lebo, and S. Skiena. Large scale online text analysis using lydia. In *Paper presented at the annual meeting of the American Political Science Association*, 2010.

S. Krause, H. Li, H. Uszkoreit, and F. Xu. Large-scale learning of relation-extraction rules with distant supervision from the web. In *Proceedings of the International Semantic Web Conference*, 2012.

J. Kunegis, A. Lommatzsch, and C. Bauckhage. The Slashdot zoo: mining a social network with negative edges. In *Proceedings of the international conference on World Wide Web*, 2009.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

S. H. Lee, P.-J. Kim, Y.-Y. Ahn, and H. Jeong. Googling Social Interactions: Web Search Engine Based Social Network Construction. *PLoS ONE*, 5(7), 2010.

A. Lehrer. *Semantic Fields and Lexical Structure*. North-Holland, 1974.

J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010a.

J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the international conference on World Wide Web*, 2010b.

Y. Li, Y. Zhang, D. Li, X. Tong, J. Wang, N. Zuo, Y. Wang, W. Xu, G. Chen, and J. Guo. PRIS at Knowledge Base Population 2013. In *Proceedings of the Text Analysis Conference*, 2013.

D. C. Liu and J. Nocedal. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Program.*, 45, 1989.

R. Malouf. A Comparison of Algorithms for Maximum Entropy Parameter Estimation. In *Proceedings of the Conference on Natural Language Learning*, 2002.

S. Maniu, B. Cautis, and T. Abdessalem. Building a signed network from interactions in wikipedia. In *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks*, 2011.

Y. Merhav, F. Mesquita, D. Barbosa, W. G. Yee, and O. Frieder. Extracting information networks from the blogosphere. *ACM Transactions on the Web*, 6(3), 2012.

C. Mieville. *The Scar*. Tandem Library, 2002.

C. Mieville. *Perdido Street Station*. Random House Publishing Group, 2003.

C. Mieville. *Iron Council*. Del Rey, 2004.

S. Milgram. The Small World Problem. *Psychology Today*, 2, 1967.

B. Min and R. Grishman. Challenges in the Knowledge Base Population Slot Filling Task. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2012.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the Annual Meeting of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP*, 2009.

G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446, 2007.

M. Peake. *The Illustrated Gormenghast Trilogy*. Vintage, 2011.

K. Peterson, M. Hohensee, and F. Xia. Email formality in the workplace: a case study on the enron corpus. In *Proceedings of the Workshop on Languages in Social Media*, 2011.

S. Petrov, D. Das, and R. McDonald. A Universal Part-of-Speech Tagset. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2012.

L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Conference on Computational Natural Language Learning*, 2009.

S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, 2010.

H. Ryu, M. Lease, and N. Woodward. Finding and exploring memes in social media. In *HT*, 2012.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communication of the ACM*, 18(11), 1975.

N. Smith. Text-driven forecasting. Technical report, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2010.

T. Strzalkowski, G. A. Broadwell, J. Stromer-Galley, S. Shaikh, S. Taylor, and N. Webb. Modeling sociocultural phenomena in discourse. In *Proceedings of the International Conference on Computational Linguistics*, 2010.

H. Takamura, T. Inui, and M. Okumura. Extracting semantic orientations of words using spin model. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 2005.

J. Turian, L. Ratinov, Y. Bengio, and D. Roth. A preliminary evaluation of word representations for named-entity recognition. In *Proceedings of the NIPS Workshop on Grammar Induction, Representation of Language and Language Learning*, 2009.

J. Turian, L. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2010.

P. D. Turney. Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002.

P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4), 2003.

M. Van De Camp and A. Van Den Bosch. The socialist network. *Decision Support Systems*, 53(4), 2012.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684), 1998.

J. Wiebe, R. Bruce, M. Bell, M. Martin, and T. Wilson. A corpus study of evaluative and speculative language. In *Proceedings of the SIGDial Workshop on Discourse and Dialogue*, 2001.

B. Yang, W. Cheung, and J. Liu. Community mining from signed social networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(10), 2007.

L. Zhu. Computational political science literature survey. Technical report, College of Information Sciences and Technology, The Pennsylvania State University, 2010.