TÉCNICO
LISBOA

Linking Entities to Wikipedia Documents

João Tiago Luís dos Santos

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Examination Committee

Chairperson: Prof. Mário Jorge Gaspar da Silva
Supervisor: Prof. Bruno Emanuel da Graça Martins
Member of the Committee: Prof.ª Maria Luisa Torres Ribeiro Marques da Silva Coheur

October 2013

Abstract

With the increasing availability of large volumes of textual information on the Web, information extraction techniques have been gaining a growing interest. In the context of my MSc thesis, I addressed the challenging information extraction problem of linking named entities, referenced within textual documents, to entries in a knowledge base, a task which has been addressed in the past, although mostly in the context of documents written in the English language. The main objective of my research work was therefore to study entity linking techniques for the case of other languages, aiming at the development of a system for associating named entities, present in Portuguese and Spanish texts, to the corresponding Wikipedia pages. Another specific entity linking problem that I addressed through this work concerns with the disambiguation of place references in text. Previous entity linking studies have argued that entities corresponding to geographic locations are particularly challenging to disambiguate, and I proposed a set of features specifically aiming to improve results on this type of entities. This dissertation formalizes the entity linking problem, describes the most relevant related work, and presents a detailed description of an entity linking system that was developed during the course of this work. It also presents an extensive experimental validation that has been carried out with different configurations of the system, using data from Wikipedia, a previously proposed dataset named XLEL-21, datasets from the TAC-KBP challenge, and two well known datasets for evaluating place reference resolution, named LGL and SpatialML. The results from experiments quantify the impact of the main ideas that were proposed through my work.

Keywords: Entity Linking , Named Entity Disambiguation , Information Extraction

Resumo

Com a crescente disponibilização de informação textual na Internet, as técnicas de extração de informação têm vindo a ganhar um maior interesse. No contexto da minha tese de mestrado, abordei um problema de extração de informação, o qual se relaciona com o ligar entidades referenciadas em documentos textuais a entradas numa base de conhecimento. Esta tarefa particular foi já abordada no passado, mas maioritariamente no contexto de documentos escritos na língua Inglesa. O principal objetivo do meu trabalho de investigação foi então o estudar técnicas de desambiguação de entidades para o caso de outras línguas, com o propósito de desenvolver um sistema que tem como objetivo associar entidades mencionadas, em textos escritos em Português e em Espanhol, às páginas da Wikipédia correspondentes. Outro sub-problema específico da tarefa de desambiguação de entidades mencionadas, que abordei neste trabalho, preocupa-se com a desambiguação de locais referidos em textos. Estudos anteriores em desambiguação de entidades argumentaram que as entidades correspondentes a localizações geográficas são particularmente difíceis de desambiguar, e eu propus um conjunto de caracteristicas que apontam especificamente a melhorar os resultados neste tipo de entidades. Concretamente, esta dissertação formaliza o problema da desambiguação de entidades, descreve o trabalho relacionado mais relevante, e apresenta uma descrição detalhada de um sistema de desambiguação de entidades que foi desenvolvido durante o percurso deste trabalho. O documento apresenta ainda uma extensiva validação experimental com diferentes configurações do sistema, usando dados da Wikipédia, um dataset previamente proposto chamado XLEL-21, dados do desafio TAC-KBP, e dois datasets bem conhecidos para avaliar a resolução de referências do tipo local, chamados LGL e SpatialML. Os resultados destas experiências quantificaram o impacto das principais ideias propostas ao longo do meu trabalho.

Keywords: Desambiguação de Entidades Mencionadas , Extracção de Informação

Acknowledgements

Tenho muito a agradecer a várias pessoas pela ajuda imprescindível ao longo do tempo em que trabalhei na minha tese de mestrado. Gostaria de começar por agradecer ao meu orientador Professor Bruno Martins, e ao meu *co-orientador*, Ivo Anastácio, pela sua ajuda, apoio e inacreditável disponibilidade, sem os quais esta fase final do curso não teria sido possível.

Gostaria ainda de agradecer o suporte financeiro da Fundação para a Ciência e Tecnologia (FCT), através de uma bolsa no projecto UTA-Est/MAI/0006/2009 (REACTION).

Por fim, deixo também um especial agradecimento a todos os que me acompanharam ao longo deste percurso, quer os que estiveram directamente envolvidos durante o processo, mas também os que sem estarem directamente envolvidos no meu percurso acadeḿico, me apoiaram incondicionalmente ao longo desta aventura.

Contents

Abstract i

Resumo iii

Acknowledgements v

1 Introduction 1

1.1 Hypothesis and Methodology 3

1.2 Contributions 4

1.3 Document Organization 6

2 Concepts and Related Work 7

2.1 The Entity Linking Problem 7

2.1.1 Wikipedia as a Knowledge Base 9

2.1.2 General Architecture of an Entity Linking System 10

2.2 Related Work 11

2.2.1 Unsupervised Approaches 12

2.2.2 Supervised Approaches 13

2.2.3 Approaches for Place Reference Resolution 21

2.3 Summary 23

3 An Entity Linking System Based on Learning From a Rich Feature Set 27

3.1 Overview 28

3.1.1 Query Expansion 28

3.1.2 Candidate Generation 29

3.1.3 Candidate Ranking 29

3.1.4 Candidate Validation 30

3.1.5 NIL Entity Resolution 30

3.2 The Considered Features 30

3.2.1 Main Contributions in Terms of the Feature Set 38

3.3 Portuguese and Spanish Resources 39

3.3.1 Named Entity Recognition 39

3.3.2 Filtering the Knowledge Base 42

3.3.3 Automatic Generation of Training Data 42

3.4 Improving the Set of Disambiguation Candidates 44

3.4.1 The Min-Hash Algorithm and Locality Sensitive Hashing 45

3.5 An Online Entity Linking Service 47

3.6 Summary 47

4 Validation Experiments 51

4.1 Evaluating the Proposed Geographic Features 52

4.1.1 Datasets and Evaluation Metrics 52

4.1.2 Experimental Results 53

4.2 Evaluating the System for the Portuguese and Spanish Languages 57

4.2.1 Comparing Different Learning Algorithms 60

4.2.2 Impact of Different Candidate Selection Strategies 62

4.2.3 Evaluating the Contribution of Different Features 64

4.3 Summary 66

5 Conclusions and Future Work 69

5.1 Main Contributions 69

5.2 Future Work 71

Bibliography 73

List of Tables

3.1 Description of the NER datasets. 40

3.2 Cross-validation results for the Portuguese and Spanish NER models. 41

3.3 Statistical characterization of the KBs used in this work. 42

4.4 Number of geo-referenced place references in the considered evaluation datasets. 53

4.5 The obtained results with the four different distance-based evaluation methodologies. 54

4.6 Results obtained with and without the proposed set of geospatial features. 55

4.7 Place references with the highest average distance, or with the highest occurrence frequency, in the LGL and SpatialML datasets. 56

4.8 Number of query entities in the different evaluation datasets that were considered. 59

4.9 Number of entries considered in each knowledge base. 59

4.10 Results for the different L2R algorithms. 60

4.11 Detailed results for the best configuration. 61

4.12 Results for the models trained with the English TAC-KBP data. 62

4.13 Number of candidate misses. 63

4.14 Results for different configurations of the candidate retrieval module. 64

4.15 Accuracy after removing sets of features. 65

4.16 Accuracy after adding sets of features to a baseline entity linking system. 66

List of Figures

2.1 General architecture of an entity linking system 10

2.2 Graph-based Entity Disambiguation, adapted from Guo *et al.* (2011) 12

3.3 Data entry form for the named entity resolution system. 48

3.4 Output generated by the named entity resolution system. 48

4.5 Geospatial distances towards the correct results for different document sizes. 57

4.6 Geospatial distances towards the correct result for different types of places. 58

Chapter 1

Introduction

With the increasing amount of textual information that is available throughout the Web, Information Extraction (IE) techniques have grown in interest. These techniques are being commonly used to process unstructured information from the Web, generally with the objective of building structured knowledge bases from the available information. In brief, we have that IE is an area of research that involves Natural Language Processing (NLP) over unstructured or semi-structured documents, in order to generate structured information. This information can be used in numerous applications, with examples being the population of knowledge bases, or the offering of support to more advanced natural language processing and information retrieval applications, such as question answering. IE can be further divided into several sub-tasks, and the most relevant sub-tasks for this work 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 *the election of **Barack Obama** is an historic event*, the string ***Barack Obama*** should be recognized as a named entity, and labeled with the type person.

- **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 ***Barack Obama*** and ***President Obama*** 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 ***Obama** was born in **Hawai***, a relation between the named entities ***Obama*** and ***Hawai*** should be identified, and perhaps later categorized as a *born in* relation, between a person and a location.

My research focused on the second problem, named entity disambiguation, also commonly referred to as **Entity Linking (EL)**. Entity linking was introduced as a specific task in the Text Analysis Conference (TAC), as a part of the Knowledge Base Population (TAC-KBP) challenge. Since then, research in automated techniques to perform this task has been growing, with numerous solutions presented in the TAC-KBP competition each year (Ji & Grishman, 2011).

There are two general types of approaches to perform named entity disambiguation, namely *corpus-based* and *knowledge-based* methods. In the first approach, the task is performed without the help of a knowledge base, and entities within the same document, or across multiple documents, are clustered together, with each cluster corresponding to a single real world entity. *Knowledge-based* approaches take advantage of large scale knowledge bases such as Wikipedia[1] or FreeBase[2], linking each entity reference to its correspondent knowledge base entry (e.g., linking entities to the corresponding Wikipedia pages). Other categorizations for the proposed methods are also possible – see Chapter 2 of this dissertation.

My work addressed the study and development of *knowledge-based* techniques for linking named entities in Portuguese or Spanish texts to the corresponding Wikipedia pages. Although the problem of Entity Linking is getting increasingly popular in the Information Extraction community, the solutions presented until now mainly focus on the English language, using the datasets made available in the context of the TAC-KBP evaluation campaign. Therefore, the main objective of this work was to develop an application to the Portuguese and Spanish languages, using adapted versions of the datasets and of the evaluation methodology from the TAC-KBP challenge. Another relevant aspect that my work addressed was the study of techniques to better disambiguate place references present in textual documents. One of the main conclusions that previous entity linking studies have reported is that geographic references constitute the most challenging type (i.e., the one that had the worst results in previous TAC-KBP competitions). Thus, I gave a particular emphasis to the study of possible approaches and specific features to better disambiguate this particular type of named entities.

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

[2] http://www.freebase.com/

1.1 Hypothesis and Methodology

My MSc research effectively tried to prove, through controlled experiments with a prototype system, two main hypothesis related to the development of entity linking systems, namely:

- The entity linking task can be addressed successfully in the case of documents written in Portuguese and in Spanish, using similar techniques to those that have been experimented in the context of the TAC-KBP evaluation event, which mostly focuses in documents written in the English language. There are some specific challenges associated to these languages (e.g., it is necessary to adapt the methods associated with the entity recognition stage, and we require the existence of a knowledge base specific to these languages that can support the disambiguation), but the same features and learning algorithms can also be used in these specific domains;
- The introduction of some additional features, based on geospatial information and on the automatic geocoding of documents, can help in the specific case of disambiguating entities corresponding to location names.

In order to evaluate the aforementioned hypothesis, I introduced some some extension to an existing entity linking system, in order to support the realization of experiments. Using this system and common metrics from the area of information extraction (e.g., the accuracy of the disambiguations, or the mean reciprocal rank), I performed experiments with various datasets:

- The SpatialML (Mani *et al.*, 2008) and LGL (Lieberman *et al.*, 2010) datasets, in English, which were already used in the past to evaluate methods specifically designed to the disambiguation of locations;
- Data for the Portuguese and Spanish languages, related with the disambiguation of entities of the type person, derived from the XLEL-21[1] collection. This collection was originally proposed in the context of a task in cross-language entity linking;
- Data derived from the Portuguese and Spanish versions of Wikipedia, where named entities are frequently linked, by Wikipedia human editors, to the corresponding pages;
- Data from the Spanish and English languages of the 2013 edition of the TAC-KBP joint evaluation on entity linking.

From the extension of the existing entity linking system, and also from the extensive set of experiments that were performed with the system, a series of important contributions were achieved, which are described in the following section.

[1] http://hltcoe.jhu.edu/datasets/

1.2 Contributions

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

- I created and evaluated Named Entity Recognition (NER) models for the Portuguese and Spanish languages, using the StanforNER[1] software framework, in order to identify named entities present in a document written in any of these two languages, prior to attempting their disambiguation. The datasets used in the training of these models were the CINTIL[2] corpus of modern Portuguese, for the case of the Portuguese language, and the dataset from CoNNL-2002[3], for the Spanish language. These models were tested using a 10-fold cross-validation technique. For the Portuguese language, an average 76.0% F_1 was achieved. As for the Spanish language, the trained models got an average 79.0% F_1.

- I developed an heuristic method with the objective of automatically generating training and test data, that could support the development of entity linking models, from Wikipedia. This method uses Wikipedia hyperlinks present in an article to generate queries, with the respective ground truths (i.e., the correct KB entry corresponding to the entity). By generating a set of queries that are correctly disambiguated, models can be trained to perform the entity linking task. These trained models can be tested with a new set of automatically generated queries as well.

- I introduced a set of geographic features, complementing the previous baseline features of the entity linking system, in order to try to improve the specific case of the disambiguation of location entities. I tested the impact of these features with specific datasets that have been used before for the evaluation of place reference disambiguation systems, namely the Local-Global Lexicon (LGL) dataset introduced by Lieberman *et al.* (2010), and the SpatialML dataset available from the Linguistic Data Consortium (Mani *et al.*, 2008). The tests showed that models trained with the geographic features that I have proposed, marginally improve the disambiguation accuracy for place references in some situations, but the differences are minimal.

- I studied and implemented a technique based on minwise hashing for filtering and retrieving possible candidate disambiguations. This technique allows one to compute the Jaccard similarity between the sets of n-grams that represent documents, in a scalable and reliable manner. This algorithm can be used to filter existing candidates through the computation of the similarity between candidates and the target document, and it is used with the support of a Locality Sensitive Hashing (LSH) technique to retrieve possible new candidates.

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

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

[3] http://www.cnts.ua.ac.be/conll2002/

Using this approach, and according to the tests that were latter performed, I managed to marginally decrease the number of candidate misses (i.e., the number of times that the correct candidate exists, but is not considered in the ranking process). Moreover, my tests also showed that the removal of candidates that had nothing to to with the reference to be disambiguated did not work using this method, as the candidate misses grew in number. This method is also used in the geocoding of new documents, a process that is essential to the computation of some of the geospatial features, by retrieving similar documents from the KB and then interpolating their coordinates.

- Also in order to try to decrease the number of candidate misses, I evaluated the use of a dataset provided by Google, containing hypertext anchors from all the links pointing to specific Wikipedia pages (Spitkovsky & Chang, 2012). Using this dataset, it was possible to complement the set of candidates with the top links associated to query references. The results showed that the use of this dataset indeed improves the candidate generation step, resolving many system errors originated by candidate misses.
- I studied possible approaches for filtering a large knowledge base like Wikipedia, in order to select only those pages that correspond to entities, i.e., persons, organizations and locations. Since the objective of the system is to perform entity linking, and since Wikipedia is not exclusively composed by entities, using Wikipedia as a knowledge base would introduce a significant amount of noise when generating possible candidates. The solution that I finally adopted involves using the structured information provided by the DBPedia[1] project, to build a consistent KB containing only entities.
- I performed several tests with documents in Portuguese and Spanish languages, in order to find out which was the best configuration for the system, and in order to quantify the system's accuracy. These tests involved the use of different groups of features, different ranking and validation algorithms, and different datasets. The datasets used in these tests were XLEL-21, the dataset provided for the TAC-KBP competition, and a dataset that I created by using hyperlinks in Wikipedia pages. These tests, and the respective datasets, will be explained in detail over Chapter 4. Overall, I discovered that that the system performs entity linking for both languages with a relatively high accuracy. The best configuration of the system, using the Wikipedia datasets, achieved an average accuracy of 97.9% for the Portuguese language, and of 98.0% for the Spanish language.
- Finally, I created an online service to perform entity linking over any given textual document. This service is available at `http://dmir.inesc-id.pt/entity-linking/`, and it allows anyone to see how a document given as input can be fully disambiguated. This online

[1] `http://dbpedia.org/`

service has been presented in the context of a demonstration in the 2013 International Conference on Web Information Systems Engineering (Santos *et al.*, 2013).

1.3 Document Organization

The rest of this dissertation is organized as follows: Chapter 2 describes the entity linking problem and also the usual architecture of a *knowledge-based* entity linking system, with basis on the related literature. It also presents related work addressing the entity linking task, namely works that used unsupervised methods, others that rely on supervised approaches, and works that have specifically focused on the disambiguation of place references. Chapter 3 presents the entity linking system that I developed through the extension of a previously existing system, giving an overview of its architecture and describing in detail every major contribution that I have given to improve the system. Chapter 4 presents the experimental validation of the thesis proposal, separately describing experiments that measured the system's accuracy when performing the disambiguation of place references, and when performing entity linking over Portuguese or Spanish texts. Finally, Chapter 5 presents the main conclusions drawn from this research, and discusses possible directions for future work.

Chapter 2

Concepts and Related Work

In this chapter, the main concepts necessary to understand the work that I have made, during the development of my MSc thesis, are presented. I start by giving a detailed description of the entity linking problem, and then I summarize some of the most relevant related work that addressed this same problem. The chapter ends with a brief overview on previous works that have specifically focused on the disambiguation of place references.

2.1 The Entity Linking Problem

Entity Linking refers to the challenging problem of mapping each named entity referenced in a textual document to the corresponding Knowledge Base (KB) entry, or determining if one such entry does not exist in the KB (Ji & Grishman, 2011). The named entities to be linked must have been previously identified by a Named Entity Recognition (NER) system, a problem which in itself constitutes another important sub-area of research within the general area of Information Extraction (Nadeau & Sekine, 2007; Whitelaw *et al.*, 2008).

The two main challenges that must be addressed by an entity linking system are name ambiguity and name variations for the entity:

- **Name Ambiguity** refers to the problem that a single string for a name might be shared by more than one entity, and thus different entities can have the same name (e.g., a mention to *Bush* might refer to the president of the United States, named *George Bush*, or to the popular rock band that is also named *Bush*).
- **Name Variations** refers to the problem that the same entity can be referred to by more than one string, i.e., an entity can have multiple distinct mentions. These alternative names, or

variations, can be acronyms (e.g. *United Nations* and *UN*), aliases (e.g. *Bradley Pitt* and *Brad Pitt*), longer entity mentions (e.g. *Bush* and *George Bush*), alternate spellings (e.g. *Osama Bin Laden* and *Usama Bin Laden*), etc.

Entity Linking must find the correct knowledge base entries to some query entities (i.e., the entities to be disambiguated), despite these two problems. To better understand the chalenges involved in entity linking, consider yet another example corresponding to the two following sentences, where the word *Armstrong* was identified as a named entity:

1. *Armstrong* joined the NASA Astronaut Corps in 1962.

2. *Armstrong* was a foundational influence in jazz.

In the previous examples, the reference *Armstrong* refers to two different entities, namely *Neil Armstrong* in the first sentence, the famous former american astronaut and the first person to set foot upon the Moon, and to *Louis Armstrong* in the second sentence, one of the biggest names in Jazz music. Through the analysis of the context in which each named entity appears, an entity linking system should assign two different identifiers, corresponding to two different entries in a given knowledge base (i.e., the entry for *Neil Armstrong* in the first sentence, and the entry for *Louis Armstrong* in the second sentence).

The entity linking task was included in the Text Analysis Conference as part of the Knowledge Base Population (TAC-KBP) track, which is a competition that promotes research in order to expand a structured knowledge base through automatic information extraction techniques. The entity linking task has become very popular within TAC-KBP, with the development of many different approaches every year (Ji & Grishman, 2011).

As described in the TAC-KBP 2011 task description, participating systems get as input a query that consists in a name string, corresponding to the target entity, and the background document in which the query appears. The query name string can refer to one of three types of entities, namely person (PER), organization (ORG) or geo-political (GPE). The system should return the correct knowledge base entry corresponding to the entity, in the context of the background document. If this knowledge base entry does not exist, a NIL should be returned as output. In addition, in the more recent editions of the competition, entity linking systems should also cluster together queries corresponding to a NIL that refer to the same real world entity.

An entity linking system has many possible uses, for instance (a) in supporting different information retrieval and natural language processing applications, like question answering tools, (b) to populate documents with links to authoritative web pages, (c) to help in areas like topic detection and tracking or in machine translation, and (d) to assisting in the clustering of web search results retrieved for queries corresponding to ambiguous entities.

2.1.1 Wikipedia as a Knowledge Base

The knowledge base that is most used for supporting entity linking tasks, and the one considered in TAC-KBP, is Wikipedia. Wikipedia has some features that can be very helpful for the task, like the descriptions associated with each entry, that can be used to help in the disambiguation process by comparing the context in which an entity appears against the context of the Wikipedia entry description. The articles also have a title that formally names the entity, which sometimes is followed by a string that discriminates entities that share the same name (e.g., *Python (programming language)* and *Python (mythology)* correspond to two different entities in Wikipedia). The Wikipedia structure also offers other helpful features, like redirection pages, disambiguation pages, infoboxes, categories, and hyperlinks, which can assist entity linking:

- **Redirect Pages:** For every alternative name that can be used to refer to an entity in Wikipedia, there is a redirect page. *United States* is an example of an entity containing multiple redirect pages. Wikipedia has entries that correspond to many different name variations (e.g., *USA*, *US*, etc.) that redirect to the main entry (in this case, *United States*).

- **Disambiguation Pages:** Ambiguous entities in Wikipedia, i.e., entities that share the same name string, are represented in the format disambiguation pages. In these pages, all entries that have the same name are listed, with the links to their own individual entry page and together with a small description.

- **Infoboxes:** An infobox is a table that presents the main facts regarding an article. There are several templates that standardize which information is most relevant to include in articles of a given subject. Infoboxes aggregate information in a semi-structured way.

- **Categories:** Every Wikipedia article has at least one category associated to it. These categories allow the articles to be placed into topics.

- **Hyperlinks:** In every Wikipedia article, when other named entity mentions appear on them, the first mention of each entity (at least) is linked to the corresponding entry in Wikipedia. Thus, the collection of Wikipedia articles constitutes, in itself, a dataset where named entity disambiguation has been performed manually by the editors of Wikipedia.

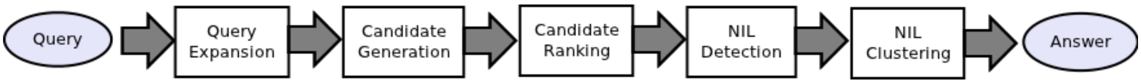


Figure 2.1: General architecture of an entity linking system

2.1.2 General Architecture of an Entity Linking System

Typically, the general architecture of an entity linking system, based on the related literature, consists in five major steps, namely (1) query expansion, (2) candidate generation, (3) candidate ranking, (4) candidate validation (i.e., NIL detection) and, (5) NIL resolution (i.e., NIL clustering). This architecture is presented in Figure 2.1. The following subsections detail each of the modules from the general architecture.

2.1.2.1 Query Expansion

An entity present in the knowledge base might be referenced in the background document by several alternative names. The main objective of this step is to expand the input query using Wikipedia structure mining (Ji & Grishman, 2011), or applying techniques that identify other names that reference the same entity (acronyms, aliases, longer entity mentions, alternate spellings, etc.) in the background document. As a result from this step, the query will now also include a set of alternative names that correspond to the entity, in addition to the name string, in the document, of the entity mention itself.

2.1.2.2 Candidate Generation

In this second step, the objective is to find possible knowledge base entries that might correspond to the named entity in the query. This can be achieved by using a variety of similarity measures (e.g., character-based string similarity metrics (Cohen *et al.*, 2003)) between the named entity from the query and all the knowledge base entries. The top-κ entries that are most likely to correspond to the entity are returned as candidates. The set of name variations found in the previous step is very important. Considering a large set of candidates can result in noisy information being added to the system, but it can also reduce significantly the number of candidate misses. The main objective of this step is to collect a limited set of likely candidate entries that will latter be analyzed in detail, during the ranking step.

2.1.2.3 Candidate Ranking

The set of candidates that result from the Candidate Generation step is examined, with basis on a predetermined set of ranking features, and sorted in a way in which the knowledge base entry that is more likely to be the correct disambiguation should stay on the top position. Most approaches use either Learning to Rank (L2R) algorithms (Liu, 2009), heuristic methods, or graph based approaches (Guo *et al.*, 2011).

2.1.2.4 Candidate Validation

The top ranked candidate, resulting from the ranking step, is evaluated in order to find if we are in the presence of a result that corresponds to the non-existence of the entity in the knowledge base, i.e., an error based on low confidence at matching. If it is decided that the candidate is not the correct entry, we assume that the correct entry does not exist in the KB and, therefore, a NIL is returned. This detection can be achieved by training a specific classification model, or by setting a threshold in the score produced by the ranking model used to sort the candidates.

2.1.2.5 NIL Entity Resolution

The objective of the NIL entity resolution step is to guarantee the disambiguation of NIL entities, by clustering together entity references that correspond to the same entity. Each cluster should have a unique identifier. This step was introduced as a requirement for entity linking systems in the TAC-KBP competition from 2011, extending the entity linking task presented in previous years. The simplest approach that can be used for NIL entity resolution consists in grouping together entities that are referenced through the same name, although other approaches have also been attempted in the competition.

2.2 Related Work

This section presents some of the most relevant related work that addressed the entity linking problem. The studied systems follow one of two types of approaches, namely unsupervised methods or supervised methods. The following subsections present these previous works in detail, further dividing supervised methods into candidate ranking approaches, document-level approaches, and cross-document approaches. This section also presents previous systems that have specifically addressed the disambiguation of place references.

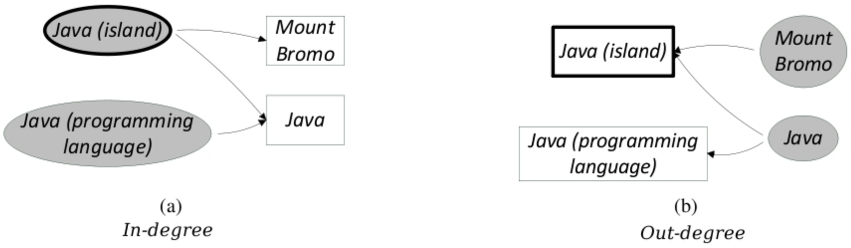


Figure 2.2: Graph-based Entity Disambiguation, adapted from Guo *et al.* (2011)

2.2.1 Unsupervised Approaches

Guo *et al.* presented an unsupervised method to perform entity linking with basis on graphs, where the set of nodes represents both the candidate KB entries and the named entities present in the context where the entity to be disambiguated appears (Guo *et al.*, 2011). Their work used degree-based measures of graph connectivity, namely out-degree and in-degree measures, to determine the importance of each candidate node, in order to assign the most important node to the query entity. Two types of nodes are considered, namely name nodes and article nodes. Name nodes refer to the name of the entity, while article nodes refer to the Wikipedia article corresponding to the entity.

Graphs based on the out-degree measure have the candidates as article nodes and the context entity mentions as name nodes. When a name string from a name node appears in the candidate article, it is assigned a direct edge from this article node to correspondent name node. As a result, the article node with the highest number of edges linked to the context name nodes (i.e., the article node with the highest out-degree) is considered the most important. The graph for this measure is illustrated in Figure 2.2(a). In this example, the entity to be disambiguated is *Java*. The considered candidates are *Java (island)* and *Java (programming language)*, and the context mentions are *Mount Bromo* and *Java*.

In-degree based graphs correspond to the opposite situation, where candidates are the name nodes and context mentions are the article nodes. An edge is defined from a context mention article to the name node of a candidate entry when that candidate name appears on the description of the article. The candidate name node with the highest in-degree is considered the most important. Figure 2.2(b) shows an example graph for the in-degree measure.

The importance of a candidate entry can also be computed through the combination of the two degree measures. When a query entity has the maximum candidate out-degree and in-degree as zero, a NIL is returned. Otherwise, the most important candidate is assigned.

Experimental results on the TAC-KBP benchmark datasets show that this method performs comparably to state-of-the-art approaches, achieving an accuracy of 0.8240 with the in-degree approach on the KBP-2010 dataset.

A different unsupervised proposal by Sarmento *et al.* did not require the existence of an external knowledge repository. These authors focused on the development of an approach capable of operating at a Web-scale (Sarmento *et al.*, 2009). Their approach relies on each mention being represented by a feature vector corresponding to the TF-IDF scores (Manning *et al.*, 2008) of remaining entity mentions in the same document. These vectors are then grouped by name (e.g., entity references with the same name are grouped together) and a graph-based clustering algorithm is applied to each group. The clustering algorithm computes pairwise similarities and, when such values are above a given threshold, it creates an edge between the corresponding entity mentions. The clusters are directly obtained by looking at the connected components. In order to avoid comparing all pairs against each other, their algorithm stops after creating an estimated sufficient number of edges to allow retrieving the same connected components, as if we were in the presence of the complete graph. Their performance results showed that name co-occurrence information was not sufficient for merging distinct facets of the same entity. This became obvious after manually inspecting the results for dominant entities, which had multiple clusters, each representing a single facet of an entity. The authors also experimented with increasingly larger samples of data and noticed that, as the size of the dataset got bigger, the complexity of the disambiguation task increased, since the number of entities and/or their respective scopes (i.e., the number of contexts in which the same real world entity appears) was also significantly larger.

2.2.2 Supervised Approaches

This section presents previous related work in which supervised approaches were used to address the entity linking task. These approaches are organized according to three different groups, namely (1) candidate ranking approaches, (2) document-level approaches, and (3) cross document approaches. Candidate ranking approaches are those where each single entity mention is disambiguated separately, whereas document-level and cross-document approaches perform the disambiguation of multiple entities at the same time, respectively those occurring in the same document, or those that occur across multiple documents in the collection, but that share some common features between this individual occurrences.

2.2.2.1 Candidate Ranking Approaches

A state-of-the-art method to perform entity linking is the one developed by Zhang *et al.* (2011). The authors proposed an entity linking framework that follows two main steps, namely (i) name variant resolution and (ii) name disambiguation.

The objective of the name variant resolution step is to find possible variations for each KB entry, through the use of Wikipedia sources like the titles of entity pages, disambiguation pages, redirect pages and anchor texts. With these sources for variations, the set of KB candidates for the entity mention to be disambiguated can be retrieved using string matching.

As for the name disambiguation step, a learning to rank algorithm (i.e., Ranking SVM (Joachims, 2002)) was used to rank each candidate determined in the previous step, using a base feature set composed of name and context similarity features, among others, as listed in the paper (Zhang *et al.*, 2011). The top ranked candidate is presented to a binary classifier that decides if a link to the entity mention, or a NIL, should be assigned.

This main structure is common to many different entity linking systems, but Zhang *et al.* proposed two important advancements to the task, namely a Wikipedia-LDA method to model the entity contexts (i.e., the source document and the Wikipedia pages) as probability distributions over Wikipedia categories, and an instance selection strategy to use auto-generated training data.

The **Wikipedia-LDA** model is used with the objective of measuring the similarity between mentions and KB entries, through their semantic representations in terms of Wikipedia categories. This is achieved by estimating the category distributions for the context of the entity mention, and for the KB entries. These distributions are estimated through a supervised *Latent Dirichlet Allocation* (i.e., the labeled LDA approach (Ramage *et al.*, 2009)) model, a state-of-the-art method for multi-label text classification. The labeled LDA model was chosen mainly because it has been show to perform better than other types of approaches, on collections with more semantically diverse labels, which is the case of Wikipedia categories.

The selection of which categories are to be used in the LDA model has a particular importance in the performance of the method, as some categories are not suitable to model the topics of a document. The authors studied five possible subsets of Wikipedia categories to create an efficient model, as listed bellow:

- **All**: a baseline set with all Wikipedia categories.
- **All-admin**: meta-categories for encyclopedia management in Wikipedia were removed from this set (e.g. *wikipedia lists*, *categories*, etc.).
- **Isa_all**: the authors state that, from their observations, most topical categories are in a *is-a*

relation to some other category. Therefore, only categories that have a *is-a* relation towards other categories are left in the set used for creating the model.

- **Isa_class**: categories from the *Is_all* set that have subcategories, or that have Wikipedia articles connected to them by a *is-a* relation, are considered as classes, and thus only this cases were kept is this set.

- **Isa_instance**: the authors considered that all categories from the *Isa_all* set that were not in the *Isa_class* set can approximately be regarded as instances (e.g., *Microsoft* is a category, but is also an instance of the class *Computer and Video Games Companies*). In this set, only the instance categories were considered, according to the previous definition.

The other particular feature introduced in this work is an instance selection strategy to effectively use auto-generated training data for entity linking. On a previous work, the same authors proposed a method to automatically generate training instances to train effective entity linking models, which consists in replacing an unambiguous entity mention in a document with its variations (Zhang *et al.*, 2010). With this approach, and using the KBP-10 document collection, 45.000 instances were created. However, there was a problem in the distribution of the examples generated through this approach, as only some types of training instances would be created.

Therefore, in this work, the authors proposed a better instance selection approach that gives a more balanced subset of auto-annotated instances. The SVM model, which is also used to rank the candidates, is used to select instances from a large dataset composed by previously auto-annotated instances. This initial classifier can be trained with a small part of the auto-generated instances, or with manually annotated data (e.g., the data from KBP-10). An iterative process then begins with the objective of selecting an informative, representative, and diverse batch of instances. In each iteration, a batch of instances is selected with basis on the current SVM model. This batch is then added to the set of training instances, adjusting the SVM model for a more accurate classification. The learning process is repeated until the classifier's confidence is at its maximum. This confidence can be calculated by summing the distances of an un-annotated development set to the SVM hyperplane.

Experimental results with the KBP-10 dataset show that the system using semantic knowledge (i.e., Wikipedia-LDA) performs better than a variant only using the base features. The system achieves an 85.9% accuracy using the base features together with the Wikipedia-LDA model based on the *isa_all* subset, while with the base features it only achieves 83.2% of accuracy.

As for the instance selection strategy, five approaches were tested to measure the benefits of this approach. For these tests only the base features were used. The system's evaluation showed that approaches that used the instance selection strategy performed better. The best method is the one where manually annotated data and auto-generated data are used along with the instance selection strategy (i.e., an accuracy of 85.5%).

Anastácio *et al.* (2011) presented another state-of-the art system, specifically developed to participate in the TAC-KBP 2011 challenge, that also used a candidate ranking approach. The authors proposed a supervised learning approach to rank the candidate entries for each named entity mention, as well to detect if the top ranked candidate was the correct disambiguation or not. This type of approach was also used to cluster named entities that do not have a KB entry association (i.e., the NILs). An experimental validation with the TAC-KBP 2011 dataset showed very competitive results, as the authors reported an overall accuracy of 79.3% when using the best configuration (i.e., the best group of features and learning algorithm). A detailed description of this system is given in Chapter 3. My entity linking system was developed as an extension of the system proposed by Anastácio *et al.*.

2.2.2.2 Document-Level Coherence Methods

Ratinov *et al.* recently developed an approach to participate in the TAC-KPB competition, using a previously proposed system named **GLOW** (*Global and LOcal Wikification*) together with the addition of a simple solution for cross-document co-reference resolution, in order to address the new task of NIL clustering that was introduced in TAC-KBP 2011 (Ratinov & Roth, 2011).

GLOW is a system previously developed by the same authors (Ratinov *et al.*, 2011), with the objective of performing *Disambiguation to Wikipedia* (D2W), which is a task that consists in cross-linking a set of mentions *M*, present in a document *d*, to the correspondent Wikipedia titles.

Although entity linking and D2W are similar tasks, they have some differences that do not enable the usage of GLOW directly. Therefore, the GLOW system was complemented with some changes in order to successfully support the entity linking task. These changes were introduced through the use of an architecture that comprises 3 steps, namely (i) mention identification, (ii) disambiguation, and (iii) output reconciliation.

Mention identification has the objective of identifying all mentions that might refer to the given query entity, present in the context document of the query. This goal can be achieved through the use of one of two methods:

- *Simple Query Identification* (SIQI) marks all the instances from the query entity in the document, with basis on exact string matching.

- *Named Entity Query Identification* (NEQI) finds named entities in the source document that correspond to the query entity, through approximate string matching. This method is similar to the query expansion step present in most entity linking systems, although the mentions found here are bound to specific locations in the document.

When using the NEQI approach, and after the set of possible mentions is computed, the canonical form (*CF*) for the query entity in the text is determined (i.e., the longest mention string, or, if no mentions were returned by the NER system, the form of the given query itself). This *CF* is then normalized (*NCF*), because GLOW only links expressions that appear as hyperlinks in Wikipedia. This NCF is the most *linkable* page of a set of pages obtained by comparing the *CF* with all titles, redirects and hyperlink anchors in Wikipedia. In the end, all instances of the *CF* in the source document are replaced by a *NCF*, and these instances are marked as the final set of mentions to be used in the next steps.

The **disambiguation** step consists on the direct application of the GLOW system to the result from the previous step, i.e., the normalized context document *d* and the set of mentions M=$\{m_1, m_2, ..., m_n\}$ appearing in this document. The system addresses the disambiguation problem as that of finding a many-to-one matching on a bipartite graph, in which document mentions form one partition, and Wikipedia articles form the other partition. The *ranker* of GLOW is executed in order to obtain the best disambiguation for each mention given as input. GLOW also has a component, called *linker*, that determines if it is better to assign a NIL, by determining if this decision improves an objective function Γ^* (which is a function that assigns higher scores to titles with a content similar to the input document, taking into account the estimated pairwise relatedness between titles). However, this component is not considered for the entity linking task, leaving this decision for the next step. The final output for this step is an N-tuple $\Gamma = (t_1, t_2, ..., t_n)$, where each t_i is the disambiguation for the entity mention m_i.

The main objective of the **output reconciliation** step is to examine the set of tuples $\{(m_i, t_i, r_i, l_i)\}$ returned by GLOW, and decide which KB entry corresponds to the query entity or, if it does not exist, assign it to the NIL result. The output resulting from GLOW has two confidence scores, r_i and l_i. The first corresponds to the ranking confidence for the title t_i being the best disambiguation for m_i, while l_i is the linker score, that gives information about if it is preferable to assign a NIL instead of the title attributed. The first stage of this step is to determine which link assigned to the set of mentions given to GLOW is the correct link (or NIL) for the entity mention from the query. This is accomplished through the usage of one of the following strategies:

- **MaxNoThres**: consists in the selection of the title with the maximum ranker confidence.
- **MaxWithThres**: similar to MaxNoThres, but if GLOW assigned a negative linker score, then the title is not considered.
- **SumNoThres**: similar to MaxNoThres, except that the ranker scores are summarized for

all mentions assigned to the same title.

- **SumWithThres**: similar to SumNoThres, and to MaxWithThres, but having titles with a negative linker score are removed from the decision.

As for the second stage of this step, the link that was found is converted to the corresponding TAC-KBP KB entry.

The experimental results in the TAC-KBP 2011 dataset showed that the micro-average accuracy of the system, using the NEQI method, is of 0.787 (using the MaxNoThres solution), which is better than when using SIQI (0.752). Results also showed that all approaches proposed to perform the output reconciliation step achieve similar results.

Another approach that worked at the document level was the one developed by Cucerzan, which performs named entity disambiguation by making use of the Wikipedia collection (Cucerzan, 2007). The author explores the use of contextual and category information, extracted from numerous distinct entities present in the Wikipedia collection, taking advantage of Wikipedia's interlinked structure. The approach aims to disambiguate an entity mention by using all entity mentions identified in the document, i.e., it considers the context for each mention to be the remaining entity mentions given in it.

The objective of Cucerzan's approach is to compare a vectorial representation of the document with the candidate vectors, in order to find the best disambiguation for the entity mentions present in the document. The *document vector* contains the information about all possible disambiguations for all entity mentions present in it, i.e., it is composed by their category tags, and by the number of occurrences of each mention occurring in the document. The candidate's vectors contain similar information, but based on the information provided by the Wikipedia entry itself. In the end, one chooses the assignment of entities to mentions that maximizes the similarity between the document vector and the candidate vectors.

The author choose not to normalize the vectors in order to account with the frequency that an entity mention is used to refer to different entities, as well as with the importance of an entity, i.e., entities that have longer articles, more category tags, and that are more frequently mentioned in other articles, should be preferred.

Although this process is based in the assumption that all instances for the entity mention in a given document only have one meaning, there are still some cases in which the *one sense per discourse* (Gale *et al.*, 1992) does not hold. In these cases, where there is not a dominant disambiguation for a mention at the document level, the problem is addressed through an iterative approach that decreases the context size. The disambiguation is performed at the paragraph level, and, if it is necessary, at the sentence level, instead of for the entire document.

The author evaluated the system in two ways. The first consisted in using a set of Wikipedia articles, by comparing references created by human contributors with the system output. The second used a set of news stories and the evaluation of the systems output was done posthoc, i.e., by seeing if the top disambiguation represented the best Wikipedia article that would satisfy the user's need for information. An accuracy of 88.3% was achieved using the first test method. Using the second method, the proposed system achieved an accuracy of 91.4%.

2.2.2.3 Cross-Document Methods

The entity linking problem can be seen from numerous points of view. Considering the addition of the NIL clustering to the task in TAC-KBP 2011, Monahan *et al.* developed a cross-document approach that does not address the problem through the usual *deductive* approach, where all entity mentions are linked to either a KB identifier or a NIL, and then the NILs are clustered. Instead, they addressed the task through an *inductive* approach, that sees the entity linking problem as particular case of cross-document coreference with entity linking (Monahan *et al.*, 2011). This approach links the entity mentions present in the documents to a knowledge base ID, or assigns them to a NIL, and then produces clusters for all entities, whether they have a knowledge base entry assignment or a NIL assignment.

To address this inductive view of the entity linking task, a multi-stage clustering algorithm, which the authors named *Four-Stage Coreferencing*, was used. The algorithm comprises four stages:

1. Group mentions according to the normalized name;

2. Resolve polysemy through supervised agglomerative clustering;

3. Resolve synonymy by merging clusters;

4. Resolve KB links for each merged cluster.

Before the execution of the previous algorithm, their approach has a step that pre-processes each document, and where each entity mention is assigned to either a KB identifier or a NIL. This is made through an extension of their previous entity linking approach, which was developed to participate in the TAC-KBP 2010 challenge (Lehmann *et al.*, 2010). This approach divided the task into three stages, namely candidate generation, candidate ranking and NIL detection:

- *Candidate Generation*: Every possible candidate for the query entity mention is found. In order to do the mapping of the entity mention strings into possible candidate entries, five candidate generators were used, namely (1) normalized articles and redirects, (2) surface text to entity map (STEM), (3) disambiguation pages, and (4) search engine results.

- *Candidate Ranking*: The candidates resulting from the generation step are ranked in order to find the most likely KB entry. To do this, a set of features representing contextual, semantic, generation, and surface evidence is extracted. Candidates are firstly ranked through the combination of these features into a numeric score (i.e., through an heuristic approach). Candidates are then re-ranked based on a logistic regression classifier trained to detect NIL entities (i.e., a machine learning approach), where the outcome label confidence is used to perform the re-ranking of the top candidates found in the first ranking stage.

- *NIL Detection*: A binary logistic classifier is trained in order to determine if the top ranked candidate is a result of the absence of the query entity mention in the KB (i.e., a NIL).

After document pre-processing, the entities from the entire document collection are used as input to the *Four-Stage Coreferencing* algorithm.

In the first stage of the *Four-Stage Coreferencing* algorithm, the entity mentions are grouped into subsets, where each subset has all the entity mentions that share identical normalized (i.e., lowercased) text strings.

In the second stage, mentions within the subsets are clustered using a supervised agglomerative clustering algorithm, which relies on the standard pairwise model. This clustering stage addresses the problem of *polysemy* (i.e., different real world entities that share the same name) among mentions. Through this stage, mentions that share the same name are resolved into separated clusters if they represent different real world entities. The clustering is achieved using an average-linkage-between-group algorithm, with a logistic classifier for the distance function. The logistic classifier uses four types of features (24 features in total), namely (1) *entity type* features, (2) *entity linking* features (i.e., features based on the links attributed by the linker during pre-processing), (3) *term similarity* features, and (4) *local context* features.

The clusters produced in the second stage are then merged, in order to resolve *synonymy*. This is accomplished by creating a graph where the nodes are clusters, and where the edges are determined by a condition, which is defined over a set of indicator functions, in order to determine if pairs of clusters correspond to the same entity. Two indicator functions were used, namely one that compares both clusters by analysing the links attributed to the mentions in the pre-processing stage, and other that compares mentions from both clusters by checking if they are embedded in a longer common phrase. In the end, after the graph is constructed, the connected vertices (i.e., the clusters) are merged to form the final entity clusters.

In the fourth, and final, stage, the clusters produced in stage three are linked to a KB entry. This is achieved through a majority voting algorithm with random tie-breaking, where each mention present in the cluster contributes with one vote, consisting in the linker's result determined during the pre-processing stage. Remember that this linkage result was either a KB entry or a NIL.

Experimental results on the TAC-KBP benchmark datasets show that this *inductive* method performs better, when compared to the *deductive* approach. The best configuration for this method achieved an accuracy score of 86.1%.

2.2.3 Approaches for Place Reference Resolution

Previous works have also addressed disambiguation tasks focusing on specific types of entities, including place references. Similarly to the general case of named entity disambiguation, the main challenges are related to ambiguity in natural language. For instance Amitay *et al.* (2004) characterized place reference ambiguity problems according to two types, namely geo/non-geo and geo/geo. Geo/non-geo ambiguity refers to the case of place names having other, non geographic meanings (e.g., *Reading* in *England* or the country named *Turkey*). On the other hand, geo/geo ambiguity arises when two distinct places have the same name. For instance almost every major city in Europe has a sister city of the same name in the New World. The geo/non-geo ambiguity is addressed when identifying mentions to places, while geo/geo ambiguity is latter addressed while disambiguating the recognized places.

In the context of his PhD thesis, Leidner (2007) surveyed approaches for handling place references in text. He concluded that most methods rely on gazetteer matching for performing the identification, together with NLP heuristics such as default senses (i.e., disambiguation should be made to the most important referent, estimated with basis on population counts), or geographic heuristics such as the spatial minimality (e.g., disambiguation should minimize the bounding polygon that contains all candidate referents) for performing the disambiguation. Some of the geospatial features used in our system are based on those surveyed by Leidner.

Martins *et al.* (2010) experimented with the usage of hidden Markov models for the recognition of place references in textual documents, together with a disambiguation model based on SVM regression that leveraged on features also inspired on the heuristics surveyed by Leidner. The regression model captured correlations between features describing the candidate disambiguations, and the geospatial distance between these candidates and the correct interpretation for the place reference. Initial experiments showed that the SVM-regression method could achieve a performance of approximately 0.6 in terms of the F_1 metric, in the task of assigning place references to a correct gazetteer identifier.

Mani *et al.* (2008) proposed the SpatialML scheme for annotating place references in text, together with a test collection annotated in this format. These authors have also reported experimental results with a statistical ranking model for place reference disambiguation, although without presenting much details about the considered approach. Specifically, the authors report a result of 0.93 in terms of the F_1 measure for the disambiguation of the recognized references.

Lieberman *et al.* (2010) and Lieberman & Samet (2011) proposed an heuristic method for the resolution of place references in textual documents, focusing on mentions to small and highly ambiguous locations. The proposed method relies on local lexicons built automatically from regional news documents, involving three main steps, namely i) inferring local lexicons, ii) performing toponym recognition, and iii) performing toponym resolution. The inference of local lexicons is made by recognizing place names in news articles from local sources, through a simple fuzzy geotagging process which returns a set of possible interpretations for ambiguous toponyms. Toponym recognition is made through a hybrid method that focuses on achieving a high recall, and that uses parts-of-speech tags for identifying proper nouns, together with lexicons and a previously-trained named entity recognition system. Finally, toponym resolution is made through a pipeline of heuristics, capturing place prominence and geographic coherence in the interpretations. A particularly interesting contribution from this work is the LGL dataset, containing a collection of news documents from local sources, which can be used to assess the accuracy of disambiguation systems in the case of highly ambiguous place references. In subsequent work, Lieberman and Samet 2012, also proposed to address the place reference disambiguation problem through a binary classification approach relying on a large set of features, by training a random forest classifier that decides, for each candidate disambiguation, if it is correct or not. Besides place prominence, the considered features reflect aspects such as the geospatial proximity between toponyms mentioned in a given textual context, and sibling relationships between disambiguations in a geographic hierarchy, for toponyms mentioned in a given textual context (i.e., an adaptive window of textual terms surrounding the reference).

Speriosu & Baldridge (2013) noted that most previous works that addressed the place reference disambiguation task have neglected non-toponym textual contexts, eventhough spatially relevant words like *downtown* or *beach*, that are not explicit toponyms, can be strong cues for disambiguation. Previously, the connection between non-spatial words and locations has been successfully exploited in data-driven approaches to the problem of document geolocation, estimating the most likely geospatial coordinates for a given textual document (Roller *et al.*, 2012). Therefore, Speriosu and Baldridge proposed to learn resolvers that use all words in local or global document contexts, using similar methods. Essentially, the authors propose to learn a text classifier per toponym, training these models using geotagged Wikipedia articles. The authors experimented with two variations of this idea, as well as with combinations of these text-driven approaches and other heuristic methods similar to those surveyed by Leidner. The first of these variations, named toponym resolution informed by predicted document locations, uses classifiers that essentially correspond to a set of language models learned from the textual contents associated to specific regions (i.e., the authors divide the Earth's surface into a set of rectangular cells, and each cell is associated to all textual contents from the corresponding region (Roller *et al.*, 2012)). This method assigns a probability to each cell from the representation of the Earth's surface, with basis on the entire contents of the document where the place reference to be disambiguated appears. Then, this probability distribution is used to estimate the probability of a given candidate disambiguation (i.e., each candidate disambiguation is assigned the probability of the grid cell that contains it), and the disambiguation with the highest probability is finally chosen. In the second variation, named Wikipedia indirectly supervised toponym resolver, the authors learn logistic regression classifiers for each possible location and place name, using the text occurring in local context windows for the place references that occur in Wikipedia articles (i.e., windows of twenty words to each side of each toponym). For each place reference in a given text, the authors use a logistic regression classifier to select the most probable disambiguation. The authors performed experiments with three different corpora, namely with the collection that was also used in the work of Leidner (2007), the Perseus Civil War and 19th Century American collection of books written about and during the American Civil War (Smith & Crane, 2001), and a dataset containing over one million articles from the English Wikipedia. The obtained results showed that the proposed approach, based on text classifiers, is more accurate than algorithms based solely on spatial proximity or metadata.

Authors like Roller *et al.* (2012) or Dias *et al.* (2012) have investigated the automatic geocoding of entire documents, instead of disambiguating individual place references. These authors have described supervised methods based on language modeling that, using the raw document text as evidence together with binned representations of the Earth's surface, classify individual documents as belonging to particular regions (i.e., the bins from the representation of the Earth), and afterwards assign geospatial coordinates of latitude and longitude with basis on these results. The authors concluded that the task of identifying a single location for an entire document provides a convenient way of evaluating approaches for connecting textual documents with locations, although we can have many documents that refer to multiple locations. Nonetheless, these types of approaches can be used in the development of features that aid in the disambiguation of individual place references.

2.3 Summary

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. Section 2.1 presented the entity linking task in detail, and discussed the main challenges that it addresses.

Specifically, we have seen that:

- Entity linking consists in mapping a named entity present in a textual document to the corresponding KB entry. If the entry that corresponds to the target named entity does not exist in the KB, a NIL should be assigned to that entity.

- Besides assigning entities to KB identifiers, NILs that correspond to the same real world entity should be grouped in a cluster that represents that entity.

- The two main challenges of the entity linking task are name ambiguity and name variations. An entity linking system must overcome these challenges and assign the correct entry to the query reference, with basis on its context.

- The knowledge base that is most used to support entity linking systems, and the one that is considered in TAC-KBP, is Wikipedia. Wikipedia has several useful features in its structure that can be very helpful for the task, namely (1) Redirect Pages, (2) Disambiguation Pages, (3) Infoboxes, (4) Categories, and (5) Hyperlinks.

- Most entity linking systems follow a standard architecture composed by five modules, namely (1) query expansion, (2) candidate generation, (3) candidate ranking, (4) candidate validation, and (5) NIL clustering.

Section 2.2 presented previous works which are related to the subject of this MSc thesis. We have specifically seen that:

- Most entity linking works follow either supervised approaches, or unsupervised approaches. Supervised methods, where there is a set of training examples available to train machine learning models, are more popular in the development of entity linking systems, as this scheme maps well to the typical experimental setting that is used in the TAC-KBP track. Still, there are many works that follow unsupervised approaches with competitive results.

- Supervised methods can be classified according to the three following categories: (1) candidate ranking, where each single entity mention is disambiguated separately by ranking a set of possible candidates, (2) document-level, where all the entities present in the same document are disambiguated at the same time, and (3) cross-document, where entities across different documents are disambiguated together.

- Some previous studies have specifically focused on the problem of disambiguating place references, proposing specific features that can be used in this task.

- All previous works that were surveyed in this section considered only the case of interpreting entity references in English documents. Using texts in the English language is still the most popular approach when developing and testing entity linking systems, despite the fact that the TAC-KBP challenge has started to include some additional languages in the task evaluation (Chinese, Spanish, etc.). In the context of my MSc thesis, particular emphasis was given to the evaluation of entity linking methods over Portuguese and Spanish texts.

Chapter 3

An Entity Linking System Based on Learning From a Rich Feature Set

This section describes the most important contributions of my research work, which was based a set of experiments with an extended version of the entity linking system by Anastácio *et al.* (2011), which in turn performed entity linking for the English language. This system is composed by five main modules, namely a query expansion module, a candidate generation module, a candidate ranking module, a candidate validation module, and, finally, a NIL clustering module.

The following section presents an overview on system, together with a description of the most important techniques that are used in the different modules. Section 3.2 details the set of features in which the considered learning methods rely on. It also presents the main contributions that I have provided to this set of features, with the objective of improving the candidate ranking step (i.e., mainly a geographic feature set). Section 3.3 describes the adaptations that have been made in order to perform the entity linking task for the Portuguese and Spanish languages, detailing the resources and methods that were used to adapt the existing system to these two languages. Section 3.4 describes approaches that I have introduced to try to improve the candidate set, by filtering results of the candidate generation step, and by adding other candidates whose text has a high similarity towards the support document. Section 3.5 presents an online application that was developed to demonstrate entity linking over any document introduced by the user. Finally, Section 3.6 summarizes the contents of this chapter.

3.1 Overview

The entity linking prototype system developed in the context of my MSc thesis is composed by five major modules, following the usual entity linking architecture that was explained in Chapter 2. The system receives as input a query, which is composed by the name to be disambiguated and the respective support document (i.e., the context in which the entity appears). This input goes then through the query expansion module, where alternative names to the entity to be disambiguated are found, and thus, included in the query. Using this richer query, possible candidates for being the correct response are generated in the candidate generation module. These candidates are then ranked, using a previously trained machine learning model, which relies on a rich set of features. The candidate that has a higher score, after this ranking, is considered the provisional correct response to the query. In the next module, i.e. the candidate validation module, this response will be evaluated using another machine learned model to detect if it is not in fact a wrong disambiguation caused by the absence of the correct entity in the KB (i.e, a NIL). This model relies on specific validation features, in addition to the ones used in the ranking step. Finally, all the queries that were assigned to NIL's are clustered. Each NIL cluster corresponds to a real world entity, i.e. all the NIL's that correspond to the same entity need to be in the same cluster. Thus, as output, the system returns the correct disambiguation for the entity name of the query, i.e., the correct link to the entry in the KB that corresponds to the real world entity, or a NIL together with the corresponding cluster id, if one such entry does not exist in the KB. In the following subsections, the main approaches taken in each module are detailed.

3.1.1 Query Expansion

The query expansion module comprises two separate mechanisms to expand the queries. The first consists on finding **acronyms** for the entity mention to be disambiguated. This is accomplished by searching the documents for a textual pattern that corresponds to a set of capital letters, followed by the acronym inserted between parentheses (e.g. *United Nations* (UN)). This mechanism also considers acronym generation, i.e., the inclusion in the query expansion result of terms that are possible acronyms for the query, and that occur in the text. The second mechanism for query expansion has the objective of finding **longer entity mentions**, present in the source document, for the query entity (e.g. *Barack Obama* is an expansion for the query *Obama*).

3.1.2 Candidate Generation

For the candidate generation module, the system uses an approach that returns the top-κ most likely entries of being the disambiguation for the given query. In order to do this, the cosine similarity metric is computed between the query and all names from entries in the knowledge base, using character n-grams (with n ranging between 1 and 4) as the unit level, instead of the more usual approach of working with document words. This means that the more n-grams the query entity name has in common with the name for the knowledge base entry, the more likely it is that this entry will be chosen as a candidate for the next steps. I used the *Apache Lucene*[1] software library for implementing this module. In addition, more possible candidates are added to this set, according to two different approaches. The first approach adds the top candidates with highest text similarity with the support document. This is achieved through a simple Locality-Sensitive Hashing (LSH) technique, that leverages the min-hash algorithm to compress large documents into small signatures (Broder, 1997). As for the second approach, a dataset provided by Google, containing all hypertext anchors pointing to a specific Wikipedia page, was used. Using this dataset, the top Wikipedia pages associated with the query reference are added to the set of candidates.

3.1.3 Candidate Ranking

As for the candidate ranking module, I used supervised Learning to Rank (L2R) approaches as described by Liu (2009), relying on the rich set of features listed in Section 3.2. I experimented with two state-of-the-art algorithms to build the candidate ranking model, namely **Ranking SVM** (Joachims, 2002) and **LambdaMART** (Burges, 2010):

- **Ranking SVM** is a pairwise ranking method that models the ranking task as a binary classification problem, which is addressed through the formalism of Support Vector Machines (SVM). This ranking algorithm is available through the SVM*Light*[2] software library.

- **LambdaMART** is a listwise ranking approach that combines a tree-boosting optimization method named MART, with a listwise ranking model known as LambdaRank. This method uses gradient boosting to optimize a particular listwise ranking cost function (i.e., the Mean Reciprocal Rank). As in the case of Ranking SVM, I used an existing open-source implementation of the *LambdaMART* learning to rank algorithm, name RankLib[3]

[1] http://lucene.apache.org/core/

[2] http://svmlight.joachims.org/

[3] http://people.cs.umass.edu/ vdang/ranklib.html

3.1.4 Candidate Validation

For the NIL detection module and, in order to detect if the chosen candidates are not actually false matches resulting from the non existence of the query entity in the knowledge base, I used an **SVM classifier** (Cortes & Vapnik, 1995) or an ensemble method known as the **Random Forest classifier** (Breiman, 2001):

- In an **SVM classifier**, the original data is mapped into a higher-dimensional space, through the use of a kernel function, where a classification hyperplane can be naturally defined. This classifier is available through the SVM*Light* software library.

- In a **Random Forest classifier**, the classification space is partitioned in terms of multiple decision trees, which are made over the attributes of the original data. The actual model is an ensemble of such trees, where the final class is assigned by majority voting. In this case, I used the *Weka*[1] software library.

3.1.5 NIL Entity Resolution

Finally, NIL references need to be clustered together. This is achieved in the system using a simple approach, that clusters together these references, with a NIL assignment, that share the same name. In previous editions of the TAC-KBP competition, results showed that this simple approach proved to be reliable, when compared to the more complex approaches developed by the participants (Ji & Grishman, 2011). Thus, and since this task is not the main focus of my MSc thesis, I adopted this simple technique to address NIL clustering.

3.2 The Considered Features

The considered learning methods for disambiguating entity references rely on a rich set of features, which can be organized according to the following groups: (1) Popularity Features, (2) Text-based Similarity, (3) Topical Similarity, (4) Name Similarity, (5) Entity Features, (6) Geographic Features, (7) Coherence Features, and (8) Validation-Only Features.

Popularity Features were introduced in the system under the assumption that popular candidates tend to be referenced more often in texts. Therefore, these features have the objective of benefiting more popular candidates:

[1] http://www.cs.waikato.ac.nz/ml/weka/

- **PageRank Score**. The PageRank score for a given candidate, computed over a graph where the nodes correspond to the knowledge base entries (i.e., the Wikipedia pages), and where the links correspond to the occurrence of hypertext links connecting the knowledge base entries. The PageRank score represents the probability of a user randomly clicking on links arriving to a determined page p_i, and is computed as follows:

  $$\mathrm{PR}(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{\mathrm{PR}(p_j)}{\mathrm{L}(p_j)} \tag{3.1}$$

  In the formula, N is the total number of pages, $M(p_i)$ is the set of pages that link to p_i, $L(p_j)$ the number of outgoing links of page p_j, and d a damping factor representing the probability of a random graph walk to continue following links (i.e., d value usually set to 0.85). These PageRank scores capture the intuition that candidates who are linked from many other highly-linked (i.e., important) knowledge base entries should be considered as more important, and thus preferred. The software used to calculate the PageRank scores was the WebGraph[1] package from the University of Milan.

- **PageRank Rank**. The rank of the given candidate in the list of all candidates, when they are ordered according to their PageRank scores.

- **Text Length**. This feature corresponds to the length of the textual description for the candidate. It is assumed that candidates should have description lengths that are proportional to their importance and popularity.

- **Text Length Rank**. The rank of the given candidate in the list of all disambiguation candidates, when they are ordered according to their textual description lengths.

- **Number of Alternative Names**. This feature corresponds to the number of alternative names associated to the candidate in Wikipedia, under the assumption that candidates with more alternatives are also more popular.

- **Alternative Names Rank**. The rank of the given candidate in the list of all candidates, when candidates are ordered according to the corresponding number of alternative names.

Text-Based Similarity Features measure the similarity between the text where the reference of the query occurs (i.e., the reference's context) and the textual description that each candidate disambiguation contains in Wikipedia:

- **Cosine Document Similarity:** The cosine similarity, using TF-IDF weights, computed between the candidate's description and the query source text, i.e., the document where the query entity occurs.

[1] http://webgraph.di.unimi.it/

- **Cosine Near Context Similarity:** The cosine similarity, using TF-IDF weights, between the candidate's description and a window of 50 tokens surrounding all occurrences of the query in the source text.

- **Cosine Candidate Beginning Similarity**. The cosine similarity, using TF-IDF weights, between the first 150 tokens of the candidate's description and the query source text.

- **Cosine Named Entity Similarity:** The cosine similarity, using TF-IDF weights, between the candidate's textual description and the query.

- **Cosine Document Rank**. The rank of the given candidate in the list of all candidates, when candidates are ordered according the text-based similarity feature called the *cosine document similarity*.

- **Query in Candidate's Text**. This feature assumes the value of one if the query occurs in the candidate's description, and zero otherwise.

- **Candidate's Name in Source Text**. Takes the value of one if the candidate's main name occurs in the query's source text, and zero otherwise.

The set of **Topical Features** uses topic-based representations for the query's source text and for the candidate's description, which were obtained using a Latent Dirichlet Allocation (LDA) topic model built with basis on all descriptions in the knowledge base. The intuition in representing the textual documents as probabilistic distributions over topics, as opposed to bags-of-words, is to minimize the impact of vocabulary mismatches.

Latent Dirichlet Allocation is a generative probabilistic model for a textual corpus, where the documents are represented as random mixtures over latent topics, and each topic is characterized by a distribution over words (Blei *et al.*, 2003).

The LDA topic model was generated with the contents of the KB, pre-processed in order to remove stop-words and word case information. Word terms were also reduced to their corresponding stems, through stemming algorithms specific to the Portuguese, Spanish and English languages (i.e., as implemented by the Lucene software library). The actual model was built through a Gibbs sampling procedure with 200 iterations, using the most frequent word stems (i.e., those who occur, at least, in 100 documents), as implemented on the JGibbsLDA[1] software framework. The value of K (i.e., the considered number of topics) was set to 400, and it was adjusted by minimizing model's perplexity on held-out data.

The actual set of topic features is as follows:

[1] http://jgibblda.sourceforge.net/

- **Topic Vector Similarity:** The cosine similarity, computed between the vectors corresponding to the candidate and to the query's topic probabilities.

- **Topic Match:** This feature takes the value of one if the LDA topic that best characterizes the candidate's description is the same that best characterizes the source text (i.e., the one with the highest probability), and zero otherwise.

- **Topic Divergence:** The symmetrized form of the Kullback-Leibler divergence metric, computed between the candidate and the query's latent topic distributions.

- **Document's Maximum Topic Probability:** The score of the the topic with the highest probability, obtained from the query's support document.

- **Candidate's Maximum Topic Probability:** The score of the the topic with the highest probability, obtained from the candidate's textual descriptions.

Name Similarity Features capture similarities between the strings of the entity references and of the candidate's names. These features use not only the strings from the main names, but also the alternative names. In the case of the query reference, alternative names are found during the query expansion step. As for the alternative names for the candidate disambiguations, I used Wikipedia redirects to get a complete set of names that also refer to the candidate in question. Therefore, when measuring name similarity with these features, all alternative names are used, and the highest similarity score from all the name combinations is the one that is considered. There are a total of 11 features in this group:

- **Name Match:** One if the named entity is an exact match with at least one of the possible names for the specified candidate, zero otherwise.

- **Name Substring**. Takes the value of one if the entity name, or if one of the candidate's names, is a substring of the other, and zero otherwise.

- **Query Starts Candidate Name:** The value of one if at least one of the candidate's names starts with the query, and zero otherwise.

- **Query Ends Candidate Name:** The value of one if at least one of the candidate's possible names ends with the query, and zero otherwise.

- **Candidate's Name Starts Query:** This feature takes the value of one if the entity name starts with at least one of the candidate's names, and the value of zero otherwise.

- **Candidate's Name Ends Query:** This feature assumes the value of one if the entity name ends with at least one of the candidate's names, and assumes zero otherwise.

- **Common Name Words:** The maximum number of common words between the query and one of the candidate's names.

- **Levenshtein Similarity:** The string similarity based on the Levenshtein metric between the candidate's name and the query. Noticing that queries are often expanded with basis on the source document, and noticing that candidates might have alternative names associated to them, the combination with the higher similarity is used as the final value.

- **Jaccard Similarity:** The similarity, based on the Jaccard token-based metric, between the candidate's name and the query. Being C the set of tokens composing the candidate's name, and Q the set of tokens that represent the query's name, the Jaccard similarity is the size of the intersection between these two sets, divided by the size of their union:

$$\mathrm{J}(Q, C) = \frac{|Q \cap C|}{|Q \cup C|} \tag{3.2}$$

- **Jaro-Winkler Similarity:** The string similarity based on the Jaro-Winkler metric between the candidate's name, or alternative names, and the query or query expansion, whichever is higher. Jaro-Winkler is variant form the Jaro string similarity measure, using a prefix scale, in order to favor strings that share a prefix of a determined length:

$$d_w = d_j + (lp(1 - d_j)) \tag{3.3}$$

  In the previous formula, l is the length of 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.

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|Q|} + \frac{m}{|C|} + \frac{m-t}{|m|}\right) & \text{otherwise} \end{cases} \tag{3.4}$$

- **Soft Jaccard Similarity:** The highest Jaccard token-based string similarity between the candidate's names and the query, using the Levenshtein edit-distance with a threshold of 2 when matching the individual tokens.

- **Soft TF-IDF Similarity:** The highest TF-IDF token-based string similarity between the candidate's names and the query, using the Jaro-Winkler distance with a threshold of 0.9 when matching the individual tokens.

Entity Features leverage on the results from a Named Entity Recognition (NER) system, specifically Stanford NER, applied to both the query's source text and to the candidate's textual description. NER systems return not only the named entities occurring in the text but also their estimated type (i.e., person, organization, or location). The set of features is as follows:

- **Common Entities:** The number of named entities shared by both the query's source text and the candidate's textual description.
- **Jaccard Similarity Between Entities:** The Jaccard similarity metric computed between the set of named entities in the query's source text, and the set of named entities from the candidate's textual description.
- **Query Type:** The named entity type, i.e. person, organization, location, or unknown, estimated when recognizing a named entity with the same string as the query. If the query is recognized more than once, the type results from a majority vote. Each type is represented by a binary feature in the final feature vector.
- **Candidate Type:** Similar to the query type, but based on the information available in the knowledge base, as DBPedia and TAC-KBP provide information about the type of an entry.
- **Type Match:** One if the query and the candidate's types are the same, zero otherwise.

Geographic Features essentially try to capture aspects related to place prominence (i.e., important places should be preferred as disambiguations) and geographic coherence in the disambiguations (e.g., textual documents tend to mention places related among themselves). These features, inspired by previous works in the area such as those of Leidner (2007) or of Lieberman & Samet (2012), are as follows:

- **Candidate Count**. The number of times that the candidate appears also as a disambiguation candidate for other place references in the same document, or in a window of 50 tokens surrounding the reference.
- **Population Count**. This feature takes the value of a particular attribute that is commonly associated to the entries in the knowledge base, corresponding to the number of inhabitants of a given candidate place, as referenced in Wikipedia infoboxes.
- **Geospatial Area**. This feature also takes the value of a particular attribute that is commonly associated to places described in the knowledge base, corresponding to the area of the region in squared Kilometers, as referenced in Wikipedia infoboxes.
- **Common Geo Entities**. The number of place references that are shared by both the query's source text and the candidate's textual description in Wikipedia.

- **Jaccard Similarity Between Geo Entities:** The Jaccard similarity metric computed between the set of place references in the query's source text, and the set of place references from the candidate's textual description.

- **Missed Geo Entities**. The number of place references in the source text that are not mentioned in the candidate's textual description.

- **Geospatial Distance**. I used an efficient similarity search method based on a locality-sensitive technique with the support of min-hash (Broder, 1997) to assign geospatial coordinates of latitude and longitude to the entire contents of the query document, afterwards measuring the geospatial distance between the coordinates of the document and those of the candidate, using the geodetic formulae from Vincenty (1975).

- **Geospatial Containment**. I again used a similarity search method based on min-hash, but this time to assign the entire contents of the query document to a geospatial region defined over the surface of the Earth, afterwards seeing if the candidate's coordinates are contained within this geospatial region.

- **Average and Minimum Distance**. The mean, and the minimum, geospatial distance between the candidate disambiguation, and the best candidate disambiguations for other place references in the same document, computed through Vincenty's formulae. The best candidates correspond to those having the highest textual name similarity.

- **Distance to Closest Reference**. The geospatial distance between the candidate disambiguation, and the best candidate for the place reference that appears closer in the same query document. The best candidate is again that which has the highest textual name similarity. This distance feature takes the value of zero if the document contains a single place reference in its text.

- **Area of the Geometric Hull**. The area of the convex hull, and of the concave hull, obtained from the geospatial coordinates of the candidate disambiguation, and from the coordinates of the best candidates for other place references made in the same document. Best candidates are again those with the highest textual similarity. The geometric hulls are computed through the implementations available in the JTS[1] software framework.

Coherence Features, or document-level features, essentially aim to capture the document's context by analyzing all entity mentions recognized in the source document, and the respective candidates associated to these references. Many previous works addressing the entity linking

[1] http://www.vividsolutions.com/jts/JTSHome.htm

task have noted that significant improvements could be achieved by performing entity disambiguation at a document level (Cucerzan, 2007), i.e. by taking into account the disambiguations produced for other entities mentioned in the same documents, when choosing the correct disambiguation. Following a similar intuition, I considered a set of features that captures the idea that candidates who are related to many other candidate disambiguations, for entities appearing in the same document, are more likely to constitute correct disambiguations. The considered set of document-level features is as follows:

- **Candidate Winner Links:** The number of times the candidate appears linked (i.e., when we have an hypertext link in the Wikipedia page for the candidate) to the possible winner candidates of the remaining entities. When using the models, as the winner is still unknown at the time when features are being computed, it is assumed that the possible winner is the one with the highest text similarity. When training the models, the actual correct disambiguations are used.

- **Contextual PageRank:** The PageRank score computed over a document graph, where the nodes correspond to all the possible candidates for all the entities present in the document and their respective immediate neighbors (i.e., the knowledge base entries that are linked to these candidates through the existence of hypertext links), and where the links represent the existence of a hyperlink connecting these nodes.

- **Contextual PageRank Rank:** The rank of the given candidate in the list of all candidates, when they are ordered according to their contextual PageRank scores, as calculated over the document graph.

Besides the aforementioned features, and noticing that the validation module only takes as input the top ranked candidate (i.e., the best possible disambiguation), I consider some **Validation-Only Features** that result from the full set of ranking scores, and check if the score for the top-ranked candidate is significantly different from that of the remaining candidates.

- **Ranking Score:** The score of the best candidate, as given by the ranking module.

- **Mean Score:** The mean score given to the query's set of candidates.

- **Difference from Mean Score:** The difference between the best candidate's ranking score and the mean score given to the query's set of candidate disambiguations.

- **Standard Deviation:** The standard deviation in the scores given by the ranking module to the query's set of candidates.

- **Number of Standard Deviations:** The number of standard deviations separating the best candidate's ranking score from the mean score.
- **Dixon's Q Test for Outliers:** This feature is obtained through the computation of the formula $(x_1 - x_2)/(x_1 - x_{last})$, where x_1 denotes the score of the top-ranked candidate, x_2 denotes the score of the candidate ranked in second place, and x_{last} denotes the score of the candidate ranked in last. If the first candidate is different from the others, then it is likely to correspond to an outlier.

3.2.1 Main Contributions in Terms of the Feature Set

Most of the features presented in the previous section were already included in the system developed by Anastácio *et al.* (2011). In this subsection, I present my specific contributions in improving the set of features that is used by the system, which essentially focused on the set of geographic features, and on performing some adjustments to the other features.

3.2.1.1 Geographic Features

Most systems developed to participate in the TAC-KBP competition reported worst results in the disambiguation of place references, and thus this particular type of entity can be considered as the most challenging. With the objective of trying to improve named entity disambiguation in the case of locations mentioned in textual documents, I introduced a new set of features that try to capture aspects related to place prominence (i.e., important places should be preferred as disambiguations) and geographic coherence in the disambiguations (e.g., textual documents tend to mention places related among themselves). In order to compute most of these features, the support document corresponding to the reference query needs to be geocoded, i.e. geospatial coordinates of latitude and longitude need to be attributed to the entire contents of the document. This was accomplished using a minwise hashing technique to find the κ nearest neighbors, i.e. the most similar entries, having themselves geospatial coordinates, that are present in the KB – see Subsection 3.4.1. Then, using the coordinates of those entries, the geographic midpoint of the centroid coordinates is computed, by converting the polar latitude and longitude coordinates into 3D Cartesian coordinates, calculating the average of the converted coordinates, and then projecting the obtained centroid back into the surface of the Earth (Jenness, 2008). Having this information available for the support document, features like the distance between the coordinates associated to the document and the coordinates of a possible candidate, can be computed. Moreover, this coordinates allow us to assign the entire contents of the query document to a geospatial region defined over the surface of the Earth, thus enabling more features like checking if a candidate is contained in the region assigned to the document.

3.2.1.2 Improving the Remaining Sets of Features

The entity linking system presented by Anastácio *et al.* (2011) constituted a strong baseline with an extensive set of features. I have nonetheless introduced some modifications to some of the existing features, with the goal of improving system's results (i.e., systems's accuracy), and also to try improving the system's computational performance. The main changes, along with the introduction of the geospatial feature set, were as follows:

- Wikipedia redirects were included in the name similarity features. Alternative names that refer to the candidate name are now used in name similarity computations. The final similarity for a particular feature is the highest value computed with all possible candidate names.
- Entity features now use the named entities recognized by the NER over the candidate article, instead of only finding the name strings of the entities, present in the support document, in the text of the candidates.
- The set of coherence features was expanded, as the contextual PageRank features were introduced to complement the already existing winner overlap feature. The computation of the overlap feature was also modified, mainly to make it more efficient.

3.3 Portuguese and Spanish Resources

In order to perform entity linking over Portuguese and Spanish documents, I had to make some adjustments to the system developed by Anastácio *et al.* (2011). One of them was to adapt the system with the objective of performing the essential subtask of Named Entity Recognition (NER). Another problem in performing entity linking in these two languages, mainly for the case of the Portuguese language, is the lack of data to train and test the system.

3.3.1 Named Entity Recognition

Named Entity Recognition concerns with the detection of named entities (i.e., persons, organizations and locations) mentioned in textual documents. This is an utterly important task for my work, namely because of the following two aspects:

- The recognition of named entities, in the support document for the given query, allows one to better understand the context in which the reference entity appears. The entities that

	CINTIL PT	**CoNNL-2002 ES**
Tokens	726.916	380.926
Entities	026.626	026.624
Person	011.253	006.265
Organization	006.436	010.474
Location	005.837	006.936
Unknown	003.100	002.949

Table 3.1: Description of the NER datasets.

co-occur with the query entity support the computation of some of the context features that have been considered.

- In a real world scenario, the objective of an entity linking system is to receive as input any given document and return it completely disambiguated, i.e., return all the entities linked to the corresponding entry in the KB. Therefore, in order to perform entity linking over a given textual document, the first step that needs to be taken is the recognition of the entities contained in it. These entities will then be considered as the queries that are passed on to the subsequent modules of the system.

The software package used to perform named entity recognition in the system was *StanfordNER*. This package already provides trained models for the English language, but lacks models for the Portuguese and Spanish languages. Nonetheless, it provides an easy framework to train new models with data from different sources and, therefore, to train models for different languages. For the Portuguese language, 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 726.916 annotated word tokens taken from texts collected from different sources and domains. For the Spanish language, it was used a similar model trained with the Spanish dataset of the *CoNLL 2002* Shared Task on Named Entity Recognition, which contains 380.926 word tokens taken from newswire articles made available by the Spanish EFE News Agency. These datasets are characterized in Table 3.1. The datasets had to be adapted in order to be in the format accepted by *StanfordNER*, and to include the common labels that I wanted to consider in the recognition processes (i.e., `PERSON`, `ORGANIZATION`, `LOCATION`, and `UNKNOWN`). Entities that were not of one of the first three types, I considered to be `UNKNOWN`.

StanfordNER 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 well-engineered feature extractors for Named Entity Recognition. The features used in the training process are as follows:

Evaluation Metric	**Portuguese**	**Spanish**
Precision	0.7702	0.8012
Recall	0.7515	0.7783
F1	0.7601	0.7895
True Positive Entities	19655	20703
False Positive Entities	5861	5145
False Negative Entities	6868	5922

Table 3.2: Cross-validation results for the Portuguese and Spanish NER models.

- **Words:** The current word, previous word, next word, and all words (before and after the current word) within a window of two tokens, are all used as features in the NER models.

- **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 a lists containing first and last names from persons.

- **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. These features check if the current word is a part of any name in a gazetteer. I used a gazetteer containing containing a completer set of Portuguese locations.

- **Word Sequences:** Word sequences related to the current word, and also previous sequences of words were used as features.

- **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 of current, previous and next positions (e.g., previous word + current shape, or current word + next shape).

To test the models included in the system, I used a ten-fold cross-validation approach. In κ-fold cross-validation the data is partitioned into κ equal size samples. Of this κ samples, one is left out of the training process (i.e., is retained to perform the validation), and the remaining κ-1 samples are used to train a NER model. This process is repeated, using all κ parts as the validation sample. In the end, one computes average scores over the κ results, thereby producing a single estimation that represents the quality of the final model. I used κ=10 in these experiments (i.e., I divided the full training corpus in ten equal samples, used nine to train a model, and one to validate that same model). The results of this validation process are presented in Table 3.2, where we can see that the considered NER models can achieve an acceptable performance.

3.3.2 Filtering the Knowledge Base

The Knowledge Base (KB) is a fundamental component of an entity linking system. Up until now, most entity linking systems perform the task by disambiguating the named entites to Wikipedia entries (i.e., all possible entries from Wikipedia are used as the KB). However, Wikipedia is composed by much more than only entities (i.e., persons, organizations or locations). This introduces a problem related with the amount of useless entries, generating a significant amount of noise. With this problem in mind, I studied possible approaches to eliminate these undesirable entries from the KB (i.e., everything that does not fit the definition of entity). The solution that I found was to use the information provided by the DBPedia project. DBPedia is a dataset that contains structured information created with basis on Wikipedia (Mendes *et al.*, 2012). The useful information that DBPedia gives to this problem is an ontology that classifies the individual Wikipedia entries. From this ontology, I retrieved everything that was classified as a person, organization, or location, thus building an adequate KB for the problem that this work proposes to address. The previous work by Mendes *et al.* (2012) describes the automated procedure used in the creation of DBPedia, through which the individual entries are classified.

Complementing this KB created with DBPedia, all Wikipedia entries that contain geospatial coordinates (i.e., entries that are locations), and were not already in the filtered KB, were also included. Both these ideas were used to create the KBs used for the each language (i.e. English, Portuguese, and Spanish) and, in Table 3.3, I present some characterization statistics.

3.3.3 Automatic Generation of Training Data

One of the major problems in developing a system that aims to perform entity linking for the Portuguese and Spanish languages is to obtain data to train the machine learning models that will realize the task. Usually, researchers in the area have used the training data that is provided by the TAC-KBP challenge. This data is mainly destined to systems that perform entity linking for the English language, although since 2012, TAC-KBP also considered a Spanish cross-linking task (i.e., that are now datasets that can be adapted to the Spanish entity liking task). However, the problem of getting a good dataset to develop a competitive Portuguese entity linking system

Language	All	Person	Organization	Location
English	1.265.307	483.003	191.802	590.502
Portuguese	0218.697	066.718	020.696	131.283
Spanish	0310.500	103.891	017.693	188.916

Table 3.3: Statistical characterization of the KBs used in this work.

still needs to be resolved. Another dataset that I used was XLEL-21, available for a number of languages to perform cross-entity linking, including Portuguese and Spanish, although this is only a small dataset, containing entities of the type person. Therefore, this is a good dataset to test how the system behaves when disambiguating entities of the type person, but it is not enough to build good models that carry out the entity linking task in other domains. In order to overcome this problem, I investigated an approach to automatically generate datasets to train and test the system. This approach relies on Wikipedia's structure. As explained in Chapter 2, Wikipedia has many useful features that can assist in entity linking tasks. The **hyperlinks**, associated with the first mention of named entity references given in an article, are one of those features, and crucial to the process of automatically generating data to train models for named entity disambiguation. These hyperlinks link the mention to the corresponding entry in Wikipedia. Thus, the collection of Wikipedia articles constitutes, in itself, a dataset where named entity disambiguation has been performed manually by the editors of Wikipedia. We therefore have that the proposed method to generate train data, and also to generate data supporting validation of the resulting models, goes through the following steps:

1. A random Wikipedia article not occurring in the KB is picked, and checked to see if it contains ate least five links whose articles correspond to an entry in the KB, and two links that correspond to a NIL (i.e., the article exists in Wikipedia, but it is not present in the KB). It is important to notice that the KB that is considered for the task does not comprise all Wikipedia entries – see Subsection 3.5 – allowing us to include NILs in the dataset. A NIL corresponds to a mention that has an hyperlink in the Wikipedia article being analyzed, but the entry corresponding to the hyperlink does not exist in the KB.

2. When an article that meets the previous conditions is found, the queries are created using the available information. A query is composed by the entity mention that contains the hyperlink, the Wikipedia article from where the entity was retrieved (i.e., the support document), and the correct disambiguation (i.e., the hyperlink that the entity possesses).

3. The process is repeated until the desired number of articles is reached. In the training datasets that were used, queries were created using a sample of 10.000 articles. As for the validation dataset, 2.500 articles were used, with the requirement that articles present in the training set could not be chosen for the validation set.

3.4 Improving the Set of Disambiguation Candidates

This section explains what I did to address a fundamental aspect in the entity linking task, concerning with the number of candidates that are generated and passed to the ranking phase. As explained in Chapter 2, in the candidate generation module, a set of possible candidate entries is generated for the entity reference to be disambiguated. This is achieved by comparing the name of the reference and all existing names in the KB, returning the top-κ most likely entries for being the correct disambiguation to the named entity. Adding a small number of candidates to this set can increase the number of candidate misses (i.e., the number of times the correct disambiguation is actually present in the KB, but is not included in the candidate set, this way loosing all possibility of being chosen as the correct disambiguation). On the other hand, if a high number of candidates is chosen in the generation step, the problem of adding unnecessary candidates to the set (i.e. more noise) is aggravated, since many more candidates with lower name similarities will be chosen, and thus be considered in the candidate ranking step. This also implies more feature computations, degrading system computational performance. Anastácio *et al.* (2011) did some tests to get an approximation of a good value to the maximum number of candidates to be generated, and these authors reported that having a maximum of 50 candidates usually is a good choice. The main objective of this part of my work was to try to improve the candidate set in three ways:

- Add new possible candidates, through the analysis of both types of textual documents (i.e., the support document and candidate's textual descriptions). This is achieved through the use of a simple Locality-Sensitive Hashing (LSH) technique, that leverages the min-hash algorithm to compress large documents into small signatures – see Subsection 3.4.1. This approach is meant to eliminate some candidate misses, by adding candidates whose name has a small similarity with the reference name, but where nonetheless their contexts are very similar.

- Filter the candidate set in order to try to remove candidates that have nothing to do with the reference that is being disambiguated. This is accomplished by calculating an approximation of the Jaccard similarity between both documents, again through the min-hash algorithm described by Broder (1997).

- Finally, a third approach involved the use of a dataset previously released by researchers at Google, which consists of the set of all hypertext anchors pointing to a specific Wikipedia page. Whenever the source query is found over this dataset, I consider as candidates for the query the top 10 Wikipedia pages more likely to be associated to the query in the dataset, since every association has a confidence score.

3.4.1 The Min-Hash Algorithm and Locality Sensitive Hashing

The min-hash algorithm was introduced in the system with the objective of computing the similarity between pairs of documents in a scalable and reliable manner.

The naive approach to finding the most similar documents, in a database of size N, involves computing all N^2 pairwise similarities. However, this quickly becomes a bottleneck for large N. Even if the task is paralelizable, overcoming the $O(N^2)$ complexity is necessary to achieve good scalability. On the other hand, precomputing the similarities is prohibitive with regard to both space and time. Even if we merely needed a single byte to store a pair's relatedness value, we would consume terabytes of storage for a reasonably sized database, and potentially much more. This is not practically viable and thus an important goal is to avoid the quadratic effect and devise appropriate pre-processing that facilitates relatedness computations on the fly.

In this work, I considered representing documents as sets of shingles (i.e., sets of contiguous character n-grams, each assigned to a globally unique identifier), and the similarity between documents is measured through an approximation of the Jaccard coefficient between the sets, obtained through a minwise hashing (i.e., min-hash) procedure.

The min-hash technique was introduced in the seminal work of (Broder, 1997), where the authors report on a successful application to duplicate Web page detection. Given a vocabulary Ω (i.e., all shingles occurring in a document collection) and two sets, S_1 and S_2, where

$$S_1, S_2 \subseteq \Omega = \{1, 2, \ldots, D\}, \tag{3.5}$$

we have that the Jaccard similarity coefficient between the sets of shingles is given by the ratio of the size of the intersection of S_1 and S_2 to the size of their union:

$$\mathrm{J}(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{|S_1 \cap S_2|}{|S_1| + |S_2| - |S1 \cap S_2|} \tag{3.6}$$

The two sets are more similar when their Jaccard index is closer to one, and more dissimilar when their Jaccard index is closer to zero. In large datasets, efficiently computing the set sizes is often highly challenging, given that the total number of possible shingles is huge. However, suppose a random permutation π is performed on Ω, i.e.,

$$\pi : \Omega \longrightarrow \Omega, \text{ where } \Omega = \{1, 2, \ldots, D\}. \tag{3.7}$$

An elementary probability argument shows that:

$$\Pr\left(\min(\pi(S_1)) = \min(\pi(S_2))\right) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \mathrm{J}(S_1, S_2) \tag{3.8}$$

In the implementation of the minwise hashing scheme, each of the independent permutations is a hashed value, in our case taking 32 bits of storage. Each of the k independent permutations is thus associated to a hash function h that maps the members of Ω to distinct values, and for any set S we take the k values of $h_{min}(S)$, i.e. the member of S with the minimum value of $h(x)$. The set of k values is referred to as the min-hash signature of a document.

Efficient nearest neighbor search is implemented through a simple Locality-Sensitive Hashing (LSH) technique, that leverages the min-hashes to compress large documents into small signatures, and preserves the expected similarity of any pair of documents. This technique uses L different hash tables (i.e., L different MapDB[1] persistent storage units), each corresponding to an n-tuple from the min-hash signatures, that is referred here as a band. At query time, it is computed a min-hash signature for the query document and then consider any pair that hashed to the same bucket, for any of the min-hash bands, to be a candidate pair. Only the candidate pairs are checked for similarity, using the complete min-hash signatures to approximate the Jaccard similarity coefficient. This way, it can be avoided the pair-wise similarity comparison against all documents.

The outline of the proposed algorithm to try to find possible candidates according to the text similarity based on n-grams, is therefore as follows: First, a list of min-hashes are extracted from each document in the collection of examples. The min-hashes are split into bands, and hashed into the L different hash tables. At query time, when finding candidates related to a given query, I start by also computing a min-hash signature from the support document textual contents. Candidates from the knowledge base with at least one identical band are considered candidates, and their similarity towards the query document is then estimated using the available min-hashes. The candidates are sorted according to their similarity, and the top kn most similar documents are returned as candidates.

This algorithm is also used in the geocoding of documents (i.e., in the assignment of coordinates to a given textual document) in order to perform the computation of some of the geographic features that were developed. The process is very similar to that of finding possible candidates. After the top kn most similar documents are known, their coordinates, if available, are interpolated, in order to estimate the geospatial coordinates of the query document. This interpolation is based on finding the geographic midpoint from the available coordinates.

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

3.5 An Online Entity Linking Service

The entity linking system was mainly designed with the objective of processing a batch of documents, and return the correct disambiguations for the queries given as input (i.e., it was prepared to participate in the TAC-KBP entity linking challenge). However, from a user point of view, a system designed to only perform the task through this approach would be uninteresting and useless. With this motivation, I developed a web-based interface that presents the entity resolution functionality through a web-based form, through which users can input the text where entities are to be resolved. The system replies with an XML document encoding the entities occurring in the texts, together with the results for their disambiguation. A stylesheet builds a web page from this XML document, where entities in the original text appear linked to the corresponding Wikipedia page. Through tooltips, the user can quickly access overviews on the entities that were referenced in the texts (e.g., I show photos representing the entities, latitude and longitude coordinates for locations, the main names associated to the entities, etc.).

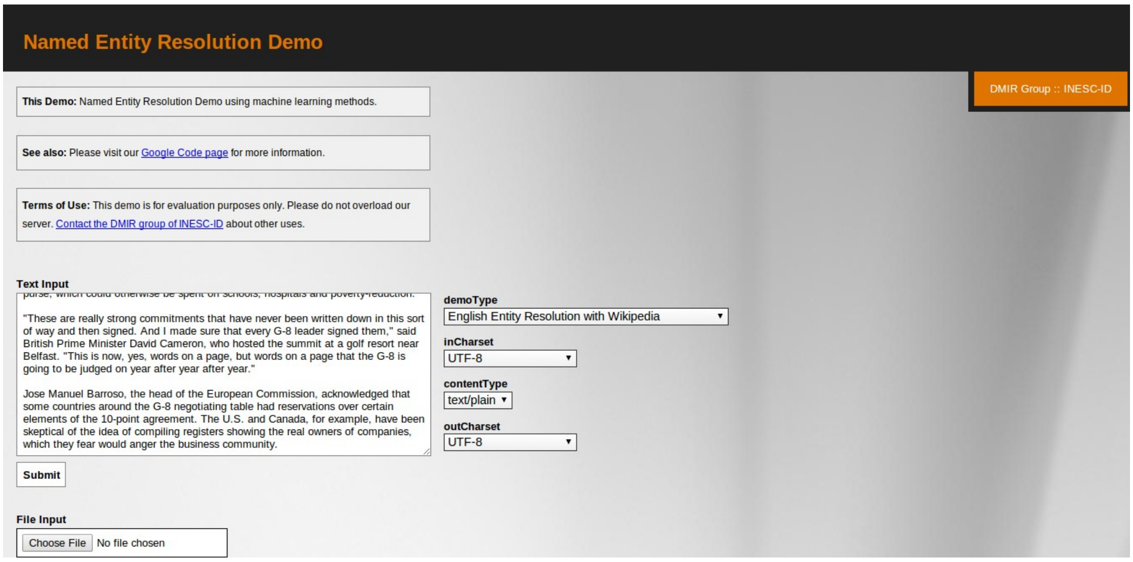
Figures 1 and 2 present two screenshots of the web-based interface for our entity recognition and disambiguation system. Figure 1 shows the main screen of the service. There are two options to introduce the desired input text to be disambiguated, namely the option to paste it in a text box, or to upload a file containing the target text. Before submitting this text to the entity linking system, the language needs to be chosen, and the system supports English, Spanish, and Portuguese.

On the other hand, Figure 2 presents the screen containing the output of the system, i.e. the result that is produced from the input textual document to be disambiguated. We can see that all entities recognized by the NER appear highlighted, and each color represents a different entity type. The entities are also linked to the corresponding Wikipedia page, according to the decision made by the system when performing entity linking. A tooltip with relevant additional information about an entity (such as the entity type, the score of the ranking algorithm, an image that represents the entity, a map location if the entity is georeferenced, etc.) pops up when the user moves the mouse over the entity.

This online service is available at `http://dmir.inesc-id.pt/entity-linking/`, and it has been presented in the context of a demonstration in the 2013 International Conference on Web Information Systems Engineering (Santos *et al.*, 2013).

3.6 Summary

This chapter presented the main contributions of my MSc research, outlining the general architecture for the entity linking system that was used and extended, and detailing the approaches



Named Entity Resolution Demo

DMIR Group :: INESC-ID

This Demo: Named Entity Resolution Demo using machine learning methods.

See also: Please visit our Google Code page for more information.

Terms of Use: This demo is for evaluation purposes only. Please do not overload our server. Contact the DMIR group of INESC-ID about other uses.

Text Input

purse, which could otherwise be spent on schools, hospitals and poverty-reduction.

"These are really strong commitments that have never been written down in this sort of way and then signed. And I made sure that every G-8 leader signed them," said British Prime Minister David Cameron, who hosted the summit at a golf resort near Belfast. "This is now, yes, words on a page, but words on a page that the G-8 is going to be judged on year after year after year."

Jose Manuel Barroso, the head of the European Commission, acknowledged that some countries around the G-8 negotiating table had reservations over certain elements of the 10-point agreement. The U.S. and Canada, for example, have been skeptical of the idea of compiling registers showing the real owners of companies, which they fear would anger the business community.

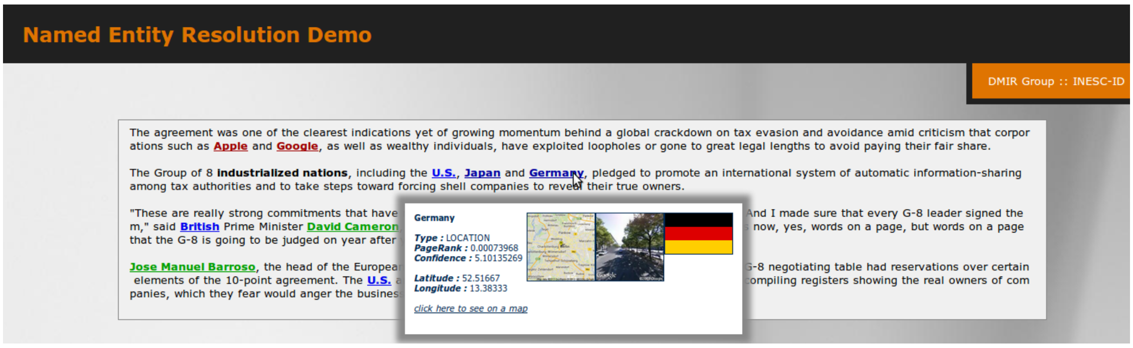
demoType: English Entity Resolution with Wikipedia ▾

inCharset: UTF-8 ▾

contentType: text/plain ▾

outCharset: UTF-8 ▾

Figure 3.3: Data entry form for the named entity resolution system.



Named Entity Resolution Demo

DMIR Group :: INESC-ID

The agreement was one of the clearest indications yet of growing momentum behind a global crackdown on tax evasion and avoidance amid criticism that corporations such as Apple and Google, as well as wealthy individuals, have exploited loopholes or gone to great legal lengths to avoid paying their fair share.

The Group of 8 **industrialized nations**, including the U.S., Japan and Germany, pledged to promote an international system of automatic information-sharing among tax authorities and to take steps toward forcing shell companies to reveal their true owners.

"These are really strong commitments that have … And I made sure that every G-8 leader signed the m," said British Prime Minister David Cameron … now, yes, words on a page, but words on a page that the G-8 is going to be judged on year after …

Jose Manuel Barroso, the head of the Europea… G-8 negotiating table had reservations over certain elements of the 10-point agreement. The U.S. a… compiling registers showing the real owners of companies, which they fear would anger the busines…

Germany

Type : LOCATION
PageRank : 0.00073968
Confidence : 5.10135269

Latitude : 52.51667
Longitude : 13.38333

click here to see on a map

Figure 3.4: Output generated by the named entity resolution system.

explored in order to extend the system to the Portuguese and Spanish languages. The chapter also explains the extensions that were used to improve the disambiguation of place references in textual documents, and also to improve the candidate generation step.

Being the main focus of my MSc thesis the development of an entity linking system capable of addressing the task for the Portuguese and Spanish languages, I specifically trained and tested named entity recognition models for both these languages, using the StanfordNER software framework. Furthermore, since the directly available data for training entity linking models in these languages is much more limited, when comparing to the data that is available for the English language, I developed a method to automatically generate training and testing data, using the source textual documents from Wikipedia. I also investigated possible methods to filter the language-specific knowledge base to be used in the entity linking task, with the objective of removing every knowledge base entry that did not correspond to an entity. This was achieved using the data provided by the DBPedia project.

I specifically created a new set of features with the motivation of trying to improve the disambiguation of locations. This geographic feature set tries to capture aspects related to place prominence (i.e., important places should be preferred as disambiguations) and geographic coherence in the disambiguations (e.g., textual documents tend to mention places related among themselves).

I also investigated new approaches to try to improve the candidate generation step, by either adding more possible candidates, in order to try to avoid system errors caused by the fact that the correct candidate is not even selected as a possible disambiguation during this step (i.e., candidate misses), or by removing candidates form the final set, with the assumption that candidates which have a fairly low text similarity, towards the support document for the respective query reference, should not be considered.

Finally, I developed an online demonstration service to perform entity linking over any given textual document. The service is prepared to perform the entity linking task for Portuguese, Spanish, and English. The service has been presented in the context of a demonstration in the 2013 International Conference on Web Information Systems Engineering.

Chapter 4

Validation Experiments

This chapter describes the experimental validation performed on the entity linking system, namely a set of experiments that measured the impact of the geospatial features in the disambiguation of place references, and a complete set of experiments that evaluated the system's performance for the Portuguese and Spanish languages.

To validate the geographic features, I used specific datasets containing only place references to be disambiguated, namely the Local-Global Lexicon (LGL) dataset and the SpatialML dataset. I also used a Wikipedia dataset created with the approach described in Subsection 3.3.3, but containing only queries of the location type. The objective of these tests is to show that the introduction of geographic features improves results when trying to disambiguate this challenging entity type. This is done through the comparison of the system with two different configurations (i.e., a configuration that uses models trained with the set of geographic features, and another that uses models that do not consider this set).

As for the second set of experiments, the main objective is to show that the system performs entity linking for the Portuguese and Spanish languages with a competitive performance. An extensive set of tests was prepared to measure the impact of different ranking algorithms and different groups of features. I also performed a study with different configurations for the candidate validation module. The datasets used for these experiments were the ones created with basis on Wikipedia with the approach presented in Chapter 3, the previously proposed XLEL-21 dataset, and also the data provided by the TAC-KBP challenge for the English and Spanish languages.

4.1 Evaluating the Proposed Geographic Features

I compared two different configurations of the system in the specific task of place reference disambiguation, namely one configuration corresponding to a standard named entity disambiguation setting, and another introducing the usage of the geospatial features described in the Chapter 3.

4.1.1 Datasets and Evaluation Metrics

For the experiments reported here, I used the approach described in Section 3.3.3 to generate the textual sources containing place references to be disambiguated (with the respective ground truths) with basis on the English Wikipedia. Besides Wikipedia, I also measured results over two previously available collections for place reference disambiguation studies, namely the Local-Global Lexicon (LGL) dataset introduced by Lieberman *et al.* (2010), and the SpatialML dataset available from the Linguistic Data Consortium (Mani *et al.*, 2008).

In brief, SpatialML considers a `PLACE` tag for annotating place references in text. There are nine different attributes defined for the `PLACE` tag, and the values of these attributes express the semantics of the annotated place references. A `LATLONG` attribute contains the geospatial coordinates corresponding to the place, encoded in a string format (e.g., numeric values for degrees, minutes, seconds, referring to latitude and longitude in the WGS-84 datum). SpatialML distinguishes between named places (e.g., names for cities, points of interest, etc.) and place nominals (e.g., *the village*, *the city*, etc.), and this study focused on the named places containing associations to geospatial coordinates, ignoring the nominals. The collection contains a total of 428 documents with 210,065 words.

As for the Local-Global Lexicon (LGL) dataset, it focuses on articles from a variety of small and geographically-distributed newspapers, thus being better-suited for evaluating place reference disambiguation on a local level and at a thinner granularity. The dataset contains a total of 588 articles with 213,446 words.

Table 4.4 presents characterization statistics for the considered datasets (i.e., the SpatialML and LGL datasets, as well as the entity disambiguation dataset built from Wikipedia itself, by using hypertext anchors from links in the Wikipedia documents as the query entities to be disambiguated). In the experiments, about 80% of the documents from the entity disambiguation dataset that was built from Wikipedia were used for training the disambiguation system, whereas the remaining 20% of the documents were used for model testing. In Table 4.4, I specifically present the total number of place references available in each collection, and also the number of place references

Dataset	Country References	City References	Other References	Total References
SpatialML	2354	1392	0698	04444
LGL	0785	2186	1491	04462
Wiki (Test)	—	—	—	12446
Wiki (Train)	—	—	—	49813

Table 4.4: Number of geo-referenced place references in the considered evaluation datasets.

of the types country and city. The remaining entities are referred to as *other*, and consider everything that is not a city or a country (e.g., states, continents, lakes, counties, etc.). As for the Knowledge Base supporting the disambiguation, it has 1,265,307 entries obtained by filtering the full set of English Wikipedia pages and keeping those those that contain geospatial coordinates, or that correspond to entities described in DBPedia as either persons, locations or organizations – see Section 3.3.2. Notice that in the case of the Wikipedia dataset, Table 4.4 only considers the non-NIL queries (i.e., in Table 4.4 we only present the number of disambiguation queries, from the Wikipedia documents, that correspond to non-NIL knowledge-based entries containing associations to geospatial coordinates, although these same documents may also contain references corresponding to NILs). When considering the NILs, we have that the Wikipedia dataset contains a total of 68,833 training references, and a total of 17,186 references for testing.

I mainly used the geospatial distance between the coordinates returned as the disambiguation and the correct geospatial coordinates, as evaluation metric. I also used accuracy (i.e., the precision at the first ranking position, which can be obtained from the ratio between the number of queries disambiguated to the correct Wikipedia entry, over the total number of queries) and the Mean Reciprocal Rank (i.e., the average of the multiplicative inverses of the ranking positions for the correct disambiguation entries) evaluation metrics, as well as the number of candidate misses (i.e., the number of times the correct candidate was not even chosen in the candidate generation step). When measuring accuracy and the MRR in the SpatialML and LGL datasets, I considered that a distance smaller than 5, 50 or 250 Kilometers corresponds to a correct disambiguation.

4.1.2 Experimental Results

Table 4.5 presents the results obtained in terms of the average and median distances, in Kilometers, between the geospatial coordinates assigned by the algorithm and the correct coordinates as given in the annotations of the different datasets.

The distance-based evaluation approach has some limitations, particularly for the case of the SpatialML and LGL datasets where we can only assess the correctness of our results by comparing geospatial coordinates. Given that distances can only be computed for knowledge base

		Models Without Geographic Features			Models With Geographic Features		
		Wikipedia	LGL	SpatialML	Wikipedia	LGL	SpatialML
Regular	Geocoded References	11865	3127	3549	11777	3167	3559
	Average Distance	0023.962	0763.137	**0136.103**	**0021.739**	**0742.040**	0139.615
	Median Distance	**0000.000**	**0002.435**	**0027.820**	**0000.000**	0002.790	0028.706
Max Distance	Geocoded References	12446	4462	4443	12446	4462	4443
	Average Distance	**0958.228**	6529.897	4140.572	1097.631	**6342.136**	**4098.593**
	Median Distance	**0000.000**	0092.734	**0054.776**	**0000.000**	**0079.896**	**0054.776**
Ranking Only	Geocoded References	12361	3442	4148	12439	4270	4379
	Average Distance	0040.558	0783.639	**0174.072**	**0040.383**	**0735.475**	0231.445
	Median Distance	**0000.000**	**0003.672**	**0054.282**	**0000.000**	0015.484	**0054.282**
Min-Hash	Geocoded References	11898	3555	3758	11777	3167	3560
	Average Distance	0033.980	0906.836	0285.790	**0021.739**	**0742.040**	**0140.871**
	Median Distance	**0000.000**	**0007.965**	**0041.439**	**0000.000**	0002.790	0054.473

Table 4.5: The obtained results with the four different distance-based evaluation methodologies.

entries associated to geospatial coordinates, we have that different strategies may result in a different number of disambiguations being used in the computation of averages (i.e., a particular configuration of the system produces more NIL results, or returns more disambiguations to knowledge base entries without coordinates, and I will not directly account with these results in the computation of average distances for the system). To address these limitations, Table 4.5 presents results for different experimental settings on what concerns the measurement of distances, namely (i) a **regular** setting where distances were only measured for those candidates to which the system assigned a non-NIL disambiguation having geospatial coordinates in Wikipedia, (ii) a **maximum distance** setting where I penalize all disambiguations made to NILs or to Wikipedia pages having no coordinates, by assigning them with a distance value of 20,038 Kilometers (i.e., half of the length of the equatorial circumference of the Earth), (iii) a setting where I only used the results from the ranking module, **ignoring NIL** classifications and measuring distances towards the coordinates of the top-ranked candidate, and (iv) a setting in which I used the **min-hash** procedure to assign geospatial coordinates to the non-NIL disambiguations that did not originally have coordinates in Wikipedia.

Table 4.6 presents the obtained results in terms of accuracy and MRR, where we can see that the accuracy across the different datasets and for both configurations remains approximately similar and reasonably high. The results in Table 4.6 were measured using the regular strategy from Table 2. The only dataset where we can calculate an exact accuracy and MRR is the Wikipedia dataset, as we have the correct disambiguation associated to each reference, not needing to measure the results with basis on distance. However, I still present the results achieved with distance based metrics for Wikipedia. As for the remaining datasets, these metrics were measured using the possible disambiguations according to the regular strategy.

The results from Tables 4.5 and 4.6 show that the system's performance benefits from the introduction of the geographic features in some situations, but it can also be observed that the

		Models Without Geographic Features			Models With Geographic Features		
		Wikipedia	LGL	SpatialML	Wikipedia	LGL	SpatialML
Exact	Total Responses	12446	–	–	12446	–	–
	Correct Responses	12027	–	–	**12074**	–	–
	Candidate Misses	5	–	–	5	–	–
	Average Accuracy	0.966	–	–	**0.971**	–	–
	Mean Reciprocal Rank	0.980	–	–	**0.983**	–	–
$\leq$ 5Km	Geocoded Responses	11865	3127	3549	11777	3167	3559
	Correct Responses	**11741**	**1734**	1499	11660	1720	1497
	Candidate Misses	3	773	1829	4	831	1836
	Average Accuracy	0.989	**0.554**	**0.420**	**0.990**	0.543	0.417
	Mean Reciprocal Rank	0.994	**0.620**	**0.441**	**0.995**	0.610	**0.441**
$\leq$ 50Km	Geocoded Responses	11865	3127	3549	11777	3167	3559
	Correct Responses	**11766**	2028	1942	11687	**2067**	**1943**
	Candidate Misses	2	371	1207	3	392	1212
	Average Accuracy	0.991	0.648	**0.544**	**0.992**	**0.653**	0.541
	Mean Reciprocal Rank	0.995	0.718	0.564	**0.996**	**0.723**	**0.570**
$\leq$ 250Km	Geocoded Responses	11865	3127	3549	11777	3167	3559
	Correct Responses	**11787**	2409	**3166**	11705	**2456**	3141
	Candidate Misses	2	92	107	3	94	118
	Average Accuracy	0.993	0.770	**0.887**	**0.994**	**0.775**	0.876
	Mean Reciprocal Rank	0.996	0.840	**0.912**	**0.997**	**0.849**	0.909

Table 4.6: Results obtained with and without the proposed set of geospatial features.

improvements are not significant. It is important to notice that the relatively high values for the geospatial distance are often due to cases such as large countries, whose centroid geospatial coordinates appear differently in Wikipedia than in the original annotations given in the SpatialML and LGL datasets. One aspect verified through the detailed analysis of the results is that the system tends to assign more NILs, when using the geographic set of features. We can see this by comparing the *regular* and the *ranking only* tests, where we can see that the number of references used rise when ignoring the NILs in the geographic models. Moreover, using geographic features the system usually performs the disambiguation to a KB entry containing coordinates, and this is why in the *minhash* test there is almost no difference when comparing with the regular test (i.e., all the disambiguations assigned in the geographic test already contained coordinates).

On what regards comparisons with the current state-of-the-art, the authors of the SpatialML dataset reported a result of 0.93 in terms of the F_1 measure for the disambiguation of the recognized geographical expressions (Mani *et al.*, 2008). On what regards the LGL dataset, the most recent study reported on a disambiguation quality of approximately 0.95 in terms of both the precision and F_1 measures (Lieberman & Samet, 2012). However, since these authors did not use Wikipedia as the knowledge base supporting the disambiguation, a direct comparison with these previous results cannot be made.

Table 4.7 shows the place names with the highest average errors, collected from the LGL (on the right) and the SpatialML (on the left) datasets, and also the references with that occur more frequently in each dataset, using the set of geographic features and the regular methodology from Table 4.5. After a careful analysis of each case, we can see that for LGL, which is a dataset containing mostly local level references, the most significant errors are related to the decision of

Name in LGL	Occurences	Average Distance
Jordan	11	10108.346 Km
Clare	03	09913.766 Km
Petersburg	01	08543.474 Km
Malta	02	08401.731 Km
Belgrade	03	08166.633 Km
U.S.	83	00000.000 Km
Georgia	62	00422.304 Km
Paris	55	07135.410 Km
Texas	52	00079.896 Km
Israel	51	00054.282 Km

Name in SpatialML	Occurences	Average Distance
Baden	1	6663.568 Km
Bristol	1	6262.896 Km
Loudoun	2	5538.943 Km
Aberdeen	1	5518.309 Km
Westwood	3	4117.233 Km
Iraq	483	0054.776 Km
Baghdad	268	0003.715 Km
Washington	096	0037.619 Km
Israel	091	0054.282 Km
Iran	090	0112.775 Km

Table 4.7: Place references with the highest average distance, or with the highest occurrence frequency, in the LGL and SpatialML datasets.

disambiguating to the most popular entry in the knowledge base. For instance, the references Jordan, Malta, Belgrade, or Paris (which often references with a high number of occurrences, but also with a high average distance in the obtained result) were disambiguated to the most popular entries that share these names (i.e., Jordan in the middle east, and Paris in France, etc.). However, all these references refer to small towns in the United States. We also have cases like Georgia, that have a considerably high average distance associated to them, despite being correctly disambiguated. This happens because, as it was mentioned before, the centroid geospatial coordinates appear differently in Wikipedia. As for the SpatialML dataset, we can also see errors resulting from choosing the most popular entry, like Aberdeen (i.e., a city in the United States, but the system disambiguated to the city in Scotland). The remaining major errors are not so clear, but may be associated with the wrong geocoding of the support document.

In a separate set of experiments, I attempted to quantify the impact that a variable such as the size of the query document has on the quality of the results. The chart in Figure 4.5 shows the distribution of the error values that were obtained for documents of different sizes, in terms of the distance towards the correct disambiguations, for the SpatialML and LGL datasets and using the regular methodology and the full set of features. The results show that there are no important differences when analyzing different documents of different sizes. The proposed method seems to be equally able to disambiguate place references in documents of different sizes.

Figure 4.6, on the other hand, shows the distribution of the error values, in terms of geospatial distances, for different types of places being referenced, in the case of the Wikipedia dataset (i.e., for the 10 place types that appear more frequently). The per-type analysis from Figure 4.6 was made through the usage of a mapping from Wikipedia categories, for the individual pages associated to the different categories, into the categories from the OpenCyc ontology, using the methodology described by Pohl (2010). The OpenCyc category system offers a better organization of concepts than that of Wikipedia, facilitating the analysis of our results on a per-type basis. We should notice that the entries that were correctly disambiguated (i.e., place references having a disambiguation error of zero Kilometers, given that in test dataset built from Wikipedia we the

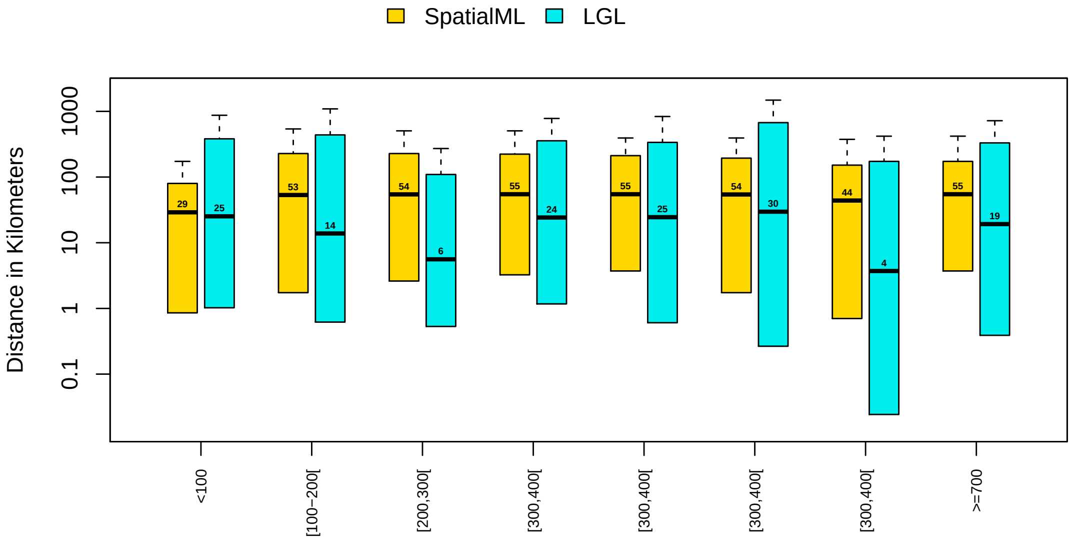


Figure 4.5: Geospatial distances towards the correct results for different document sizes.

place references assigned to the exact same geospatial coordinates that appear in the KB) are not represented in Figure 4.6 (i.e., we only show the cases corresponding to errors).

Through the analysis of the results from Figure 4.6, we can see that the different types of places appear to have a distinct distribution in terms of the errors that are produced. For instance places associated with the category *Port Cities* seem to be particularly hard to disambiguate, whereas types such as *Railway Stations* generally present small errors, in terms of geospatial distance.

4.2 Evaluating the System for the Portuguese and Spanish Languages

This section describes the experimental procedures conducted for evaluating the entity linking system for the Portuguese and Spanish languages. These experiments mainly focused on comparing different configurations of the proposed entity linking approach, using documents from recent dumps for the Portuguese and Spanish versions of Wikipedia, both as information sources (i.e., trough the automatic generation of training and test data – see Chapter 3) and as the knowledge base entries supporting the disambiguation (i.e., the filtered knowledge bases built through DBPedia), and also with the Portuguese and Spanish documents available in the XLEL-21 dataset. Moreover, in order to compare the results obtained over these two languages against

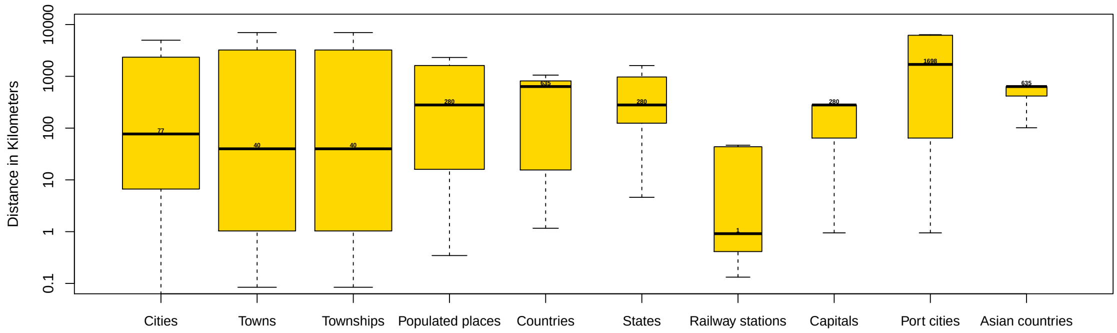


Figure 4.6: Geospatial distances towards the correct result for different types of places.

those obtained for English, and also in order to compare the performance of the proposed system against other approaches in the area that have been evaluated with English datasets, I also report on some experiments made with English Wikipedia data, and with the collection from the English entity-linking task of the Text Analysis Conference (i.e., with data from TAC-KBP 2013).

XLEL-21 was originally developed to support the training and evaluation of systems for cross-language linking of named entities (i.e., person names), from twenty-one non-English languages into an English knowledge base build from Wikipedia and equivalent to the one used in TAC-KBP. However, again with the support of DBPedia, I managed to convert this cross-language entity linking dataset, into two datasets for entity linking in the Portuguese and Spanish languages. This was achieved through the mapping of the correct disambiguations for queries in the XLEL-21 dataset, into the corresponding pages in the Spanish and Portuguese versions of Wikipedia, using the links in DBPedia between pages in different versions (i.e., languages) of Wikipedia. This method was also used for the dataset provided by TAC-KBP for the Spanish language, as this is also a cross-language entity linking task.

Noticing that the considered disambiguation features consist of either importance or similarity scores that are not directly associated to the language of the context document, I also made some experiments using models trained from the English dataset made available in the 2013 edition of the TAC-KBP event (Ji & Grishman, 2011), in tests with documents in other languages.

Table 4.9 presents characterization statistics for the considered datasets containing queries to be disambiguated (i.e., the datasets from XLEL-21, as well as entity disambiguation datasets built from Wikipedia itself, by using hypertext anchors from links in the Wikipedia documents as the query entities to be disambiguated). The considered queries correspond to named entities

Dataset	NIL	PER	ORG	GPE	ALL
XLEL-21 PT Train	62.5%	01.681	00.000	00.000	01.681
XLEL-21 PT Test	61.4%	00.443	00.000	00.000	00.443
XLEL-21 ES Train	29.9%	00.820	00.000	00.000	00.820
XLEL-21 ES Test	34.6%	00.208	00.000	00.000	00.208
Wiki PT Train	28.0%	21.129	07.805	19.216	66.888
Wiki PT Test	28.5%	04.511	01.803	05.524	16.567
Wiki ES Train	28.3%	18.722	05.927	23.182	66.697
Wiki ES Test	28.4%	04.038	01.351	06.438	16.513
Wiki EN Train	27.5%	17.426	11.418	20.865	68.570
Wiki EN Test	27.5%	04.280	02.728	05.443	17.171
KBP-13 EN Train	50.4%	03.546	05.416	03.168	12.130
KBP-13 EN Test	50.2%	00.686	00.701	00.803	02.190
KBP-13 ES Train	54.0%	01.333	01.136	01.421	03.890
KBP-13 ES Test	43.3%	00.695	00.762	00.660	02.117

Table 4.8: Number of query entities in the different evaluation datasets that were considered.

Language	KB from DBPedia	KB from TAC-KBP
Portuguese	0.218.697	121.624
Spanish	0.310.500	204.150
English	1.265.307	783.054

Table 4.9: Number of entries considered in each knowledge base.

belonging to one of three groups, namely people, organizations, and geopolitical entities. As for the Knowledge Bases supporting the disambiguation, two types of KBs were used. The first type relates with the Wikipedia KBs filtered using DBPedia, in order to keep only entries of the type person, organization or location. These KBs were used in the tests related to the Wikipedia datasets. As for the second type, it addresses the remaining datasets for these experiments, using either the entire TAC-KBP KB for the English tests, or a subset of this KB with only the entries present in the Wikipedia dump of the respective language. A characterization of these KBs is presented in Table 4.9.

Unless otherwise stated, I used accuracy (i.e., the precision at the first ranking position) to measure the disambiguation system's performance. Formally, this metric corresponds to the ratio between the number of correctly disambiguated queries (i.e., those where the disambiguation candidate that is ranked first is the correct one), divided by the total number of queries. In some particular experiments I also report ranking quality results in terms of the Mean Reciprocal Rank (MRR) metric, which corresponds to the average of the multiplicative inverses of the ranking positions for the correct disambiguations.

The following sections detail particular sets of experiments, which were designed to evaluate different aspects of the entity linking system.

Ranking Model	Dataset	Overall Accuracy	Ranking Accuracy	MRR
SVMrank	XLEL-21 PT	97.3%	**99.4%**	**99.4%**
	XLEL-21 ES	93.8%	**98.0%**	**98.0%**
	Wiki PT	97.1%	96.1%	97.7%
	Wiki ES	96.9%	95.8%	97.5%
	Wiki EN	95.8%	94.6%	96.7%
	KBP-13 EN	**80.3%**	**86.7%**	**89.9%**
	KBP-13 ES	**68.6%**	**68.1%**	**74.1%**
LambdaMART	XLEL-21 PT	**97.5%**	**99.4%**	**99.4%**
	XLEL-21 ES	**94.7%**	**98.0%**	**98.0%**
	Wiki PT	**97.9%**	**97.1%**	**98.4%**
	Wiki ES	**98.0%**	**97.3%**	**98.4%**
	Wiki EN	**97.2%**	**97.2%**	**97.7%**
	KBP-13 EN	78.3%	83.9%	87.1%
	KBP-13 ES	66.0%	66.9%	73.2%

Table 4.10: Results for the different L2R algorithms.

4.2.1 Comparing Different Learning Algorithms

In a first set of experiments, I considered the full set of features introduced in Section 3.2, and measured the impact that different ranking algorithms (i.e., Ranking SVM or ensembles of trees in the LambdaMART model) could have on the results.

Notice that listwise algorithms such as LambdaMART support the direct optimization of a given evaluation metric (e.g., P@1 or MRR). Since some evaluation metrics can be more informative to the learning algorithm than others (Yilmaz & Robertson, 2010), separate experiments were designed with models trained for optimizing the Mean Reciprocal Rank (MRR) or the P@1. However, since in the tests optimizing for P@1 consistently lead to better results, only the values achieved with LambdaMART models optimizing this particular metric are reported.

I also experimented with different validation algorithms, namely SVMs and Random Forests, but the later again consistently outperformed the former, reason why I focused the presentation of the results on the ranking algorithms, which produced more variability. All the results presented in this MSc thesis were therefore produced with a Random Forest classifier as the algorithm in the validation module. The results for this first set of experiments are presented in Table 4.10, where we can see that the accuracy across different languages and datasets remains approximately similar and reasonably high. Notice that the reported Mean Reciprocal Rank (MRR) takes only into account the non-NIL queries, as it would not make sense to measure the ranking position of the correct candidate in the case of the NIL entries.

The results show that there is no ranking model that clearly outperforms the other, although using an ensemble of LambdaMART rankers does obtain the best performance in the Wikipedia datasets. However, as a downside, this is the most time consuming algorithm of the two that were considered. Since the majority of the testes performed were made using the Wikipedia datasets,

Dataset	Module	PER	ORG	GPE	ALL
	Ranking (R)	99.4%	–	–	99.4%
XLEL-21 PT	Validation (V)	97.7%	–	–	97.7%
	R + V	97.5%	–	–	97.5%
	Ranking (R)	98.4%	97.4%	96.0%	97.1%
Wiki PT	Validation (V)	99.9%	99.9%	99.9%	99.9%
	R + V	98.3%	97.3%	95.9%	97.9%
	Ranking (R)	97.1%	–	–	97.1%
XLEL-21 ES	Validation (V)	96.6%	–	–	96.6%
	R + V	94.7%	–	–	94.7%
	Ranking (R)	99.2%	97.4%	96.0%	97.3%
Wiki ES	Validation (V)	100%	99.9%	99.9%	99.9%
	R + V	99.2%	97.3%	95.9%	98.0%

Table 4.11: Detailed results for the best configuration.

the rest of the experiments reported in this work were made with LambdaMART ensembles as the ranking algorithm.

A detailed presentation of the results obtained, for the Portuguese and Spanish languages, with the best ranking algorithm is given in Table 4.11, showing that the disambiguation of geo-political entities is particularly challenging, with the system achieving the worse results on this particular entity type, despite also having a reasonably high accuracy in all datasets. Notice that the accuracy reported for the ranking module takes only into consideration non-NIL queries, whereas the reported validation accuracy ignores the queries which were incorrectly ranked (i.e., it only accounts for errors in classifying the NILs).

In a separate experiment, I attempted to see if models trained with the English KBP-13 dataset would also work well for the case of disambiguating entities in Portuguese and Spanish texts (and consequently, if the learning-based systems that have been developed in the TAC-KBP competition would also give good results). We therefore experimented with the usage of LambdaMART ensembles trained with the dataset made available in the TAC-KBP English entity linking task, for the disambiguation of entities in Portuguese or Spanish texts. Table 4.12 presents the obtained results, showing that these models do indeed offer a good performance when ported to different languages. The reason for this may be related to the fact that the considered disambiguation features consist of either importance or similarity scores that are not directly associated to the language of the context document, and are therefore transferable to other languages.

On what regards comparisons with the current state-of-the-art for the English entity linking task, we have that the best system (i.e., an overall accuracy of 82.3%) participating in TAC-KBP 2009 was based on a TF/IDF ranking model, while the second and third systems (i.e., overall accuracies of 80.3% and 79.8%, respectively) both used learning to rank approaches, namely the ListNet algorithm with a feature set that includes attributes based on the similarity between the query mention and the entity, including named entity overlap, and SVMrank with a diverse feature

Dataset	Overall Accuracy	MRR	Ranking Accuracy	Validation Accuracy
XLEL-21 PT	92.1%	95.9%	94.2%	94.2%
XLEL-21 ES	85.1%	96.9%	95.6%	87.6%
Wiki PT	81.1%	92.4%	87.5%	89.0%
Wiki ES	81.1%	92.2%	87.8%	88.8%
Wiki EN	85.8%	93.1%	88.9%	93.3%
KBP-13 EN	78.3%	87.1%	83.9%	85.2%
KBP-13 ES	65.5%	71.9%	65.6%	81.9%

Table 4.12: Results for the models trained with the English TAC-KBP data.

set that includes cosine similarity, search engine popularity measures, and named entity counts.

In TAC-KBP 2010, the winning entry (i.e., an overall accuracy of 85.8%) was submitted by a team from the Language Computer Corporation, using an approach based on a large set of features representing contextual, semantic, and surface evidence. A binary logistic classifier was used for detecting NIL entities, and the confidence scores of this classifier were used for ranking entities.

In the 2011 edition, the winning system, developed by Cucerzan (2011), obtained an overall accuracy of 86.8%. The system employs both entity representations in context/topic spaces and statistical mappings of surface forms (strings used for mentioning entities in text) to entities, as extracted from the Wikipedia collection.

The best systems for the 2012 edition are not reported in this study, since the paper detailing the results from all systems is not available.

In comparison, the work proposed here uses a much richer set of features, but since the entity linking system was configured and tested with the data made available in the 2013 TAC-KBP edition, a direct comparison with systems form previous editions can not be made. Although, we can see that the system achieves close in performance comparing to the previous years best systems. After manually inspecting some of the produced disambiguation errors, chosen at random, I noticed that the system often overestimates the importance of the popularity features, even when contextual similarity was high with the correct referent. In a separate set of experiments, I attempted to quantify the impact of the different types of features. Also, misspelled references and acronyms tended to produce many system errors, either with candidate miss errors or with wrong NIL attributions.

4.2.2 Impact of Different Candidate Selection Strategies

A fundamental aspect in entity linking is the number of candidates that are passed to the ranking module. Selecting a high number of candidates would improve recall, lowering the number of

Dataset	Candidate Misses	% of total queries	Overall Accuracy
XLEL-21 PT	1	0.584%	97.5%
XLEL-21 ES	1	0.735%	94.7%
Wiki PT	3	0.025%	97.9%
Wiki ES	6	0.051%	98.0%
Wiki EN	15	0.121%	97.2%
KBP-13 EN	59	2.694%	78.3%
KBP-13 ES	41	1.937%	66.0%

Table 4.13: Number of candidate misses.

cases where the correct referent exists in the knowledge base but is not selected for ranking. However, more candidates also means more feature computations, as well as more noise, since candidates with lower name similarities will be considered. Our experiments showed that, in our usual system setup, selecting up to 50 candidates for each query results in a considerably low candidate miss rate, as shown in Table 4.13. A manual analysis of the results, produced for the English TAC-KBP collection, showed most candidate misses come from highly ambiguous place names (e.g., *Columbia* or *St. Louis*), followed by acronyms (e.g., *TNT*, *HDFC* or *SF*) and generic entities (e.g., *democratic party* or *public security police*).

Table 4.14 presents the accuracy of the system for different configurations of the candidate selection module, namely by considering a different maximum number of candidates, by using a candidate generation method that uses the Locality-Sensitive Hashing (LSH) technique with the support of the min-hash algorithm, and also by adding the top 10 candidates associated to the query reference in a dataset provided by Google. The results show that using the Wikipedia datasets, and retrieving the top 50 candidates according to Lucene's similarity already resulted in a fairly low number of candidate misses. However, the introduction of the remaining two approaches managed to get an even lower number of candidate misses, being the dataset form Google the most helpful tool in this process. Notice that, sometimes, the accuracy and MRR scores decrease when the number of candidate misses gets lower. This may seem odd at first, but since the system considers document level features (i.e., the coherence feature set, and some geographic features) that are directly related to all the candidates associated to all the references in the document, accuracy and MRR results may vary when changing the number of candidates assigned to each reference.

In order to try to improve the candidate generation step, I also experimented with the use of a candidate filtering step based on the Jaccard n-gram similarity between the textual contents of the candidate and those of the query document, as approximated through the min-hash procedure (Broder, 1997). However, I choose not to present an extensive evaluation of this approach, since it consistently produced much worst results in terms of the number of candidate misses,

Dataset	Min-Hash	Google	Max. Candidates	PER	ORG	GPE	All	MRR	Misses
Wiki PT	no	no	30	96.2%	94.3%	93.0%	96.0%	95.6%	357
			40	95.1%	97.0%	97.9%	97.5%	97.7%	098
			50	98.4%	97.7%	95.9%	98.0%	98.4%	010
	yes	no	30	96.3%	94.0%	92.9%	96.0%	95.6%	345
			40	97.9%	96.8%	95.1%	97.5%	97.8%	095
			50	98.5%	97.6%	95.9%	97.9%	98.4%	009
	no	yes	30	98.0%	97.2%	95.5%	97.6%	98.6%	081
			40	98.4%	97.5%	95.9%	97.9%	98.5%	024
			50	98.4%	97.8%	96.0%	98.0%	98.5%	004
	yes	yes	30	98.0%	97.0%	95.3%	97.6%	97.9%	074
			40	98.3%	97.4%	95.8%	97.9%	98.5%	022
			50	98.3%	97.3%	95.9%	97.9%	98.4%	003
Wiki ES	no	no	30	95.4%	93.8%	93.2%	95.7%	95.1%	419
			40	98.7%	96.4%	95.1%	97.5%	97.6%	108
			50	99.3%	96.9%	96.1%	98.0%	98.4%	015
	yes	no	30	93.4%	94.2%	95.4%	95.8%	95.2%	411
			40	98.6%	96.6%	95.2%	97.5%	97.7%	106
			50	99.2%	97.9%	96.1%	98.1%	98.5%	014
	no	yes	30	98.0%	96.3%	95.8%	97.6%	97.7%	102
			40	99.2%	97.0%	96.0%	98.0%	98.3%	024
			50	99.4%	97.1%	96.1%	98.1%	98.5%	007
	yes	yes	30	97.9%	96.4%	95.9%	97.6%	97.7%	097
			40	99.1%	97.0%	96.0%	98.0%	98.3%	023
			50	99.2%	97.3%	95.9%	98.0%	98.4%	006

Table 4.14: Results for different configurations of the candidate retrieval module.

and therefore also in the overall accuracy and MRR of the system. This is related to the fact that the candidate's source text and query's support text would often have a zero similarity score according to the min-hash approximation. Most times, a document may refer an entity but the overall textual content is different from the expected candidate's text. Since this approach used an approximation of the Jaccard similarity coefficient, the probability of the similarity being zero in these cases is greater then when using the Jaccard similarity itself, leading to a higher number of errors in this step.

Regarding the candidate selection process, it is also important to mention that the aforementioned simple query expansion techniques are very important to reduce the candidate misses and consequently improve system performance. In the tests performed, since the candidate generation process adopted already returned most candidates correctly, the impact of the query expansion method. Although in the English TAC-KBP dataset, the query expansion module decreases candidate misses by 70 to 59, thus improving system accuracy.

4.2.3 Evaluating the Contribution of Different Features

One important and interesting question is the contribution of the different types of features to the overall results. We would specifically like to know how important is a particular type of information to the named entity linking task. In this section, this problem is studied by removing features of a specific type to see how much they contribute to the final accuracy scores.

Portuguese Wikipedia Dataset				
Features	PER	ORG	GPE	All
All	98.3%	97.3%	95.9%	97.9%
-Name Similarity	94.8%	93.7%	94.0%	94.9%
-Text Similarity	97.5%	96.1%	94.8%	97.2%
-Entity	96.1%	93.1%	93.5%	96.4%
-LDA	98.2%	97.4%	95.9%	97.9%
-Popularity	98.3%	97.3%	95.8%	97.9%
-Document Level	98.3%	97.3%	95.9%	97.9%
-Geographic	98.2%	97.3%	95.8%	97.8%
Spanish Wikipedia Dataset				
Features	PER	ORG	GPE	All
All	99.2%	97.3%	95.9%	98.0%
-Name Similarity	96.2%	92.7%	93.7%	96.0%
-Text Similarity	98.4%	95.2%	94.2%	96.9%
-Entity	97.0%	91.5%	93.5%	94.8%
-LDA	99.2%	97.2%	96.0%	98.0%
-Popularity	99.2%	97.2%	95.9%	98.0%
-Document Level	99.1%	96.7%	95.7%	97.9%
-Geographic	99.2%	97.0%	95.8%	97.9%

Table 4.15: Accuracy after removing sets of features.

Ranking accuracy results, for the Portuguese and Spanish Wikipedia datasets, are presented in Table 4.15, using our best performing configuration (i.e., LambdaMART models for ranking, and Random Forest models for validation).

The results show that name and text similarity features, as well entity features, are the most helpful, since they present the most significant performance drops after their removal. However, the impact of the other features in the overall quality of the results was not so clear. Given the small differences in performance after each type of features is removed, it is possible that several of those features are complementary or redundant. To confirm it, further experiments were made, where instead of removing one type of feature from a complete system, I added the features from these types to a baseline system. I considered as baseline a system with just the name similarity features, which significantly outperformed all other options according to a separate experiment. Table 4.16 presents the obtained results. These results show that each group of features has its own contribution to the system, since system's performance rises when introducing each group to the baseline system. The text similarity features seem be the set that improves the results the most in either language, followed by the new document level features, and the popularity features. Also notice that the specific set of geographic features had a strong impact in place reference disambiguation, having almost no impact at all in the remaining entity types.

As a side note regarding the LDA features, I also experimented with LDA models considering

Portuguese Wikipedia Dataset				
Features	PER	ORG	GPE	All
Name Similarity	92.0%	86.7%	69.3%	84.4%
+Text Similarity	95.4%	91.9%	93.5%	94.4%
+Entity	96.1%	92.8%	79.7%	91.4%
+Popularity	94.5%	91.1%	91.2%	93.1%
+LDA	94.7%	90.4%	89.7%	92.2%
+Document Level	94.8%	91.3%	91.5%	92.9%
+Geographic	92.2%	87.2%	83.6%	89.4%
Spanish Wikipedia Dataset				
Features	PER	ORG	GPE	All
Name Similarity	92.9%	81.2%	67.2%	82.4%
+Text Similarity	97.2%	90.7%	92.8%	94.5%
+Entity	96.4%	89.7%	73.3%	87.9%
+Popularity	95.3%	86.6%	89.3%	92.1%
+LDA	95.9%	87.5%	89.0%	91.9%
+Document Level	95.5%	88.9%	92.1%	93.1%
+Geographic	93.0%	81.7%	83.8%	88.8%

Table 4.16: Accuracy after adding sets of features to a baseline entity linking system.

different values for the parameter K (i.e., for the number of topics). The particular values obtained in these experiments are not reported in this MSc thesis, but the results showed that changing the number of topics did not significantly influence the system performance. It should also be noted that the default value of $K = 400$ was adjusted by minimizing the model's perplexity on held-out data.

4.3 Summary

This chapter described the experimental validation of the techniques proposed in my MSc thesis.

In order to test the impact of the newly introduced geographic features, which were described in Section 3.3, I used a collection of Wikipedia documents with place references to be disambiguated, and also two well known datasets that have been used in many previous works, namely LGL and SpatialML. The experiments performed showed that the differences between the two configurations were minimal. For the LGL dataset, the system seems to perform better using the new set. However, with the SpatialML dataset the system presents most times better results without geographic features. In the Wikipedia dataset, results were very similar, but when using geographic features the system assigns more NILs. The most challenging aspect of this set of tests was to try to find a way to compare the two approaches fairly, as the results were measured with basis on the coordinates associated with the correct disambiguation and, therefore, the number of queries used to perform the validation, for both configurations, was different.

To test the entity linking system for the Portuguese and Spanish languages, collections of Wikipedia documents with references to be disambiguated were used for each language. Moreover, a dataset available for cross-language entity linking with 21 different languages, including Portuguese and Spanish, was also used to test the system. In addition, complementary tests were performed for the English language, using the same method to generate a Wikipedia dataset containing textual documents to be disambiguated, and using the data made available in the TAC-KBP competition.

A first set of experiments had the objective of determining which was the best ranking algorithm to perform the entity linking task, when comparing two well known ranking algorithms, namely LambdaMART and SVMRank. The results show that there is no ranking model that clearly outperforms the other, although using an ensemble of trees within a LambdaMART ranker does obtain the best performance in Wikpedia datasets used in most of the tests performed.

In another experiment, ranking and validation models were trained using the data provided by the TAC-KBP competition for the English language. We saw that using these models in the context of others languages still offer a good performance.

A set of experiments was also conducted to test different configurations of the candidate generation module, to determine which is the best configuration (i.e., the one that gets the lowest number of candidate misses). Results showed us that retrieving up to 50 candidates from Lucene, and using the dataset form Google to retrieve more possible candidates to the query reference prevents most candidate misses. As for the candidate generation using LSH with the support of the min-hash algorithm, results were not so clear, as the number of candidate misses in the considered datasets remained practically the same.

In what regards the impact of the various groups of features, results showed that removing certain sets of features, from the main set, would drop the system's performance, namely name similarity features, text similarity features, and entity features. Moreover, I made another set of experiments where each set of features was added to a baseline system containing only name similarity features. Results showed that the set of text similarity features is the one that improves the most the system's performance, along with the document-level features and popularity features. It can also be seen that the set of geographic features improves the baseline system's results in what regards the disambiguation of place references.

Chapter 5

Conclusions and Future Work

This dissertation presented the research work that was conducted in the context of my MSc thesis. I described an entity linking system capable of disambiguating named entities into Wikipedia pages, for texts written in English, Portuguese, and Spanish languages. The main approaches that were developed through this work focused on extending a previous existing entity linking system, initially designed with objective of participating in the English TAC-KBP challenge, in order to perform named entity disambiguation over textual documents written in Portuguese and Spanish. Despite the recent advances in the entity linking task, we have that few previous works have evaluated entity linking 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 relatively simple learning-based approach for named entity disambiguation, which uses a rich set of features and out-of-the-box supervised learning methods to achieve a performance in line with that of current state-of-the-art approaches, as reported on experiments focusing on the English language.

The extended system was also complemented with new features that aimed to improve certain aspects, such as the disambiguation of place references over the documents, with the global objective of increasing system's overall performance and accuracy.

5.1 Main Contributions

Through a series of experiments, I have shown that the extended entity linking system performs the named entity disambiguation task, over textual documents written in Portuguese and Spanish, with a high accuracy. Moreover, this system is in line with the current state-of-the-art, when comparing results with the best systems developed to participate in the TAC-KBP challenge. In order to perform the experiments that are reported in this dissertation, the system was modified to consider two new languages, and also to try to improve its computational performance. More specifically, my work provided the following main contributions:

- In order to address the fundamental baseline task of Named Entity Recognition (NER), I created models for the Portuguese and Spanish languages. With these models, the system became able to recognize entity mentions in textual documents written in Portuguese and Spanish, with an considerably high accuracy (i.e., an average $F1$ score of 76.0% for the Portuguese language, and of 79.0% for the Spanish language).

- I studied and developed a method to automatically create entity linking datasets, i.e., datasets that can be used to train and tests models that will carry out the entity linking task. This method takes advantage of the hyperlinks present in Wikipedia documents, to create a set of queries associated with support text and correct disambiguations (i.e., the KB entry associated with the hyperlink). This was motivated by the fact that there is little availability of data for languages other than English.

- With the objective of trying to improve the disambiguation of place references, which are known to be particularly challenging entities, I introduced a new set of features, which I called geographic features. The specific experiments that were performed to test the impact of these features, showed that they introduce marginal improvements in some cases, but do not have a strong overall impact when compared to models trained without these features. However, when adding this set of features to a simple baseline system containing only features related to name similarity, experiments showed an improvement in the system's performance for place references.

- I investigated a minwise hashing technique with the objective of computing the Jaccard similarity between the sets of n-grams that represent documents, in a scalable and reliable manner. This technique was introduced in the system with three major objectives. The first was to geocode documents (i.e., assign coordinates of latitude and longitude to textual documents), in order to compute some of the new geographic features. The second objective relates with improving the candidate set of disambiguations, by efficiently finding more candidates that might correspond to the reference being analyzed (i.e., an attempt at lowering candidate misses, by adding the top candidates whose textual descriptions are very similar to the support document). Tests showed that the number of candidate misses dropped, but not significantly. Finally, the third objective was to filter candidates with a low similarity score towards the support document, according to the minwise approximation to the Jaccard similarity, in order to eliminate possible *noise* in the candidates set. However, experimental results consistently showed that despite the number of disambiguation candidates being lower, this approach would also rise the number of candidate misses.

- I evaluated the use of a dataset provided by Google, containing hypertext anchors from all links pointing to specific Wikipedia pages, with the objective of avoiding possible candidate misses. Results showed that this dataset is indeed really helpful in the candidate generation step, allowing the system to avoid a considerable number of candidate misses.
- I developed an approach to remove entries from the Wikipedia KBs that were not considered as entities (i.e., persons, organizations, or locations), thus removing *noise* form the KB. This approach uses the structured information provided by DBPedia.
- I developed an online entity linking service, that allows one to use the system through a web-based form. This service has been presented in the context of a demonstration in the 2013 International Conference on Web Information Systems Engineering (Santos *et al.*, 2013), and it allows a user to input any given document, in either in English, Portuguese or Spanish, and see the result of the entity linking process. This service is available at `http://dmir.inesc-id.pt/entity-linking/`.
- I performed an extensive set of tests to evaluate the final entity linking system, using documents in Portuguese and Spanish. I tested different configurations of the system, variating the groups of features that ere being used, the ranking algorithms, or the different candidate generation setups. Results showed that, with the best configuration, the system achieved an accuracy of 97.9% for the Portuguese language, and of 98% for the Spanish language (i.e., when using the Wikipedia auto-generated datasets). Moreover, I also performed tests with the data made available in the TAC-KBP competition, and showed that results in these more difficult datasets remain reasonably high, and close to the state-of-the-art results reported for systems that participated in previous editions of this competition.

5.2 Future Work

Despite the interesting results, that are also still many open challenges for future work. It would be interesting, for instance, to experiment with the usage of additional features in the ranking and validation modules. I believe that features derived from structured information associated to the knowledge based entries (i.e., features derived from slot-filling methods) could provide particularly rich information for entity disambiguation purposes.

The Information Retrieval community has also started to look at the problem of relational learning to rank, explicitly considering cases in which there exists relationship between the objects to be ranked (Qin *et al.*, 2008). For future work, and noticing that entities referenced in the same context (e.g., in the same document or in documents from a same collection) should be similar to one another, it would be interesting to experiment with relational learning methods in order to explore document- or collection-level disambiguation directly at the level of the learning algorithm, going beyond the document-level features that were already considered in this work.

Finally, we have that the experiments reported in this dissertation have mostly addressed the disambiguation of named entity references, assuming the existence of a named entity recognition system. In the case of our experiments, the entities to be disambiguated were explicitly provided as queries, and we separately trained a named entity recognition model, that was used in the computation of features depending on named entities, through the Stanford NER toolkit. For future work, it would be interesting to jointly evaluate a complete approach for recognizing and disambiguating named entities in Portuguese and Spanish texts, particularly experimenting with the LEX++ unsupervised and language independent named entity recognition approach that was proposed by Downey *et al.* (2007).

Bibliography

AMITAY, E., HAR'EL, N., SIVAN, R. & SOFFER, A. (2004). Web-a-where: Geotagging web content. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

ANASTÁCIO, I., CALADO, P. & MARTINS, B. (2011). Supervised learning for linking named entities to wikipedia pages. In *Proceedings of the Text Analysis Conference*.

BLEI, D., NG, A. & JORDAN, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*.

BREIMAN, L. (2001). Random forests. *Machine Learning*.

BRODER, A.Z. (1997). On the resemblance and containment of documents. In *Proceedings of the Conference on Compression and Complexity of Sequences*.

BURGES, C.J.C. (2010). From ranknet to lambdarank to lambdamart: An overview. *Microsoft Research Technical Report*.

COHEN, W.W., RAVIKUMAR, P. & FIENBERG, S.E. (2003). A comparison of string metrics for matching names and records. In *KDD Workshop On Data Cleaning And Object Consolidation*.

CORTES, C. & VAPNIK, V. (1995). Support-vector networks. *Machine Learning*.

CUCERZAN, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

CUCERZAN, S. (2011). Tac entity linking by performing fulldocument entity extraction and disambiguation. In *Proceedings of the Text Analysis Conference*.

DIAS, D., ANASTÁCIO, I. & MARTINS, B. (2012). Geocoding Textual Documents Through Hierarchical Classifiers Based on Language Models. *Linguamática, Revista para o Processamento Automático das Línguas Ibéricas*.

DOWNEY, D., BROADHEAD, M. & ETZIONI, O. (2007). Locating complex named entities in web text. In *Proceedings of the 20th international joint conference on Artifical intelligence*.

GALE, W., CHURCH, K. & YAROWSKY, D. (1992). One sense per discourse. In *Proceedings of the MLT workshop on Speech and Natural Language*.

GUO, Y., CHE, W., LIU, T. & LI, S. (2011). A graph-based method for entity linking. In *Proceedings of the International Joint Conference on Natural Language Processing*.

JENNESS, J. (2008). Calculating areas and centroids on the sphere. In *Proceedings of the 28th Annual ESRI International User Conference*.

JI, H. & GRISHMAN, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

JOACHIMS, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*.

LEHMANN, J., MONAHAN, S., NEZDA, L., JUNG, A. & SHI, Y. (2010). LCC Approaches to Knowledge Base Population at TAC 2010. *Proceedings of the Text Analysis Conference*.

LEIDNER, J. (2007). Toponym resolution: a comparison and taxonomy of heuristics and methods.

LIEBERMAN, M. & SAMET, H. (2011). Multifaceted toponym recognition for streaming news. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

LIEBERMAN, M. & SAMET, H. (2012). Adaptive context features for toponym resolution in streaming news. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

LIEBERMAN, M., SAMET, H. & SANKARANARAYANAN, J. (2010). Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proceedings of the IEEE International Conference on Data Engineering*.

LIU, T.Y. (2009). Learning to rank for information retrieval. *Information Retrieval*.

MANI, I., HITZEMAN, J., RICHER, J., HARRIS, D., QUIMBY, R. & WELLNER, B. (2008). SpatialML annotation scheme, corpora, and tools. In *Proceedings of the International Conference on Language Resources and Evaluation*.

MANNING, C.D., RAGHAVAN, P. & SCHTZE, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

MARTINS, B., ANASTÁCIO, I. & CALADO, P. (2010). A machine learning approach for resolving place references in text. In *Procedings of the AGILE International Conference on Geographic Information Science*.

MENDES, P.N., JAKOB, M. & BIZER, C. (2012). DBpedia: A multilingual cross-domain knowledge base. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*.

MONAHAN, S., LEHMANN, J., NYBERG, T., PLYMALE, J. & JUNG, A. (2011). Cross-lingual cross-document coreference with entity linking. *Proceedings of the Text Analysis Conference*.

NADEAU, D. & SEKINE, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*.

POHL, A. (2010). Classifying the wikipedia articles into the opencyc taxonomy. In *Proceedings of the ISWC Workshop on the Web of Linked Entities*.

QIN, T., LIU, T.Y., ZHANG, X.D., WANG, D.S., XIONG, W.Y. & LI, H. (2008). Learning to rank relational objects and its application to web search. In *Proceeding of the International Conference on World Wide Web*.

RAMAGE, D., HALL, D., NALLAPATI, R. & MANNING, C.D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

RATINOV, L. & ROTH, D. (2011). Glow tac-kbp 2011 entity linking system. In *Proceedings of the Text Analysis Conference*.

RATINOV, L., ROTH, D., DOWNEY, D. & ANDERSON, M. (2011). Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

ROLLER, S., SPERIOSU, M., RALLAPALLI, S., WING, B. & BALDRIDGE, J. (2012). Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

SANTOS, J., MARTINS, B. & BATISTA, D. (2013). Document analytics through entity resolution. In *Proceedings of the International Conference on Web Information Systems Engineering*.

SARMENTO, L., KEHLENBECK, A., OLIVEIRA, E. & UNGAR, L. (2009). An approach to web-scale named-entity disambiguation. In *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition*.

SMITH, D.A. & CRANE, G. (2001). Disambiguating geographic names in a historical digital library. In *Proceedings of the European Conference on Digital Libraries*.

SPERIOSU, M. & BALDRIDGE, J. (2013). Text-driven toponym resolution using indirect supervision. In *Proceedings of the Annual Metting of the Association for Computational Linguistics*.

SPITKOVSKY, V.I. & CHANG, A.X. (2012). A cross-lingual dictionary for english wikipedia concepts. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*.

VINCENTY, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review XXIII*.

WHITELAW, C., KEHLENBECK, A., PETROVIC, N. & H. UNGAR, L. (2008). Web-scale named entity recognition. In *Proceedings of the ACM conference on Information and knowledge management*.

YILMAZ, E. & ROBERTSON, S. (2010). On the choice of effectiveness measures for learning to rank. *Information Retrieval*.

ZHANG, W., SU, J., TAN, C.L. & WANG, W. (2010). Entity linking leveraging automatically generated annotation. In *Proceedings of the International Conference on Computational Linguistics*.

ZHANG, W., SU, J. & TAN, C.L. (2011). A wikipedia-lda model for entity linking with batch size changing instance selection. In *Proceedings of International Joint Conference on Natural Language Processing*.