

Linking Entities to Wikipedia Documents

João Santos
joao.d.santos@ist.utl.pt
Instituto Superior Técnico
Av. Professor Cavaco Silva
2744-016 Porto Salvo,
Portugal

ABSTRACT

This paper addresses the challenging information extraction problem of linking named entities in text to entries in a large knowledge base such as Wikipedia. The approach, which is essentially an evolution of a system originally developed in the context of the English Entity Linking Task of the Text Analysis Conference, uses supervised learning to rank candidate knowledge base entries for each named entity, and then for classifying the top-ranked entry as the correct disambiguation or not. In this paper, I analyze the fundamental design challenges involved in the development of a learning-based entity-linking system, and provide extensive experimental results with both Portuguese and Spanish texts, for a wide range of methods and feature sets. The experiments demonstrate the effectiveness of supervised learning methods, showing that out-of-the-box algorithms and relatively simple to compute features can obtain a high accuracy in this task.

Keywords

Information Extraction, Entity Linking, Machine Learning

1. INTRODUCTION

Given the large amounts of textual data currently available on the Web, research on information extraction methods to automatically extract structured information from these sources is getting increasingly popular. Information Extraction (IE) can be further divided into several sub-problems, most notably *named entity recognition* [29], *relationship extraction* [10], and *named entity disambiguation* [15]. This work addresses the later sub-problem, also known as grounding, cross-document co-reference resolution, or named entity linking. It can be briefly summarized as the task of mapping an entity previously recognized by a Named Entity Recognition (NER) system, i.e. the reference, to an identifier specific to the concept that the named entity is referring to in the text, i.e. the referent. The named entity disambiguation problem is currently receiving substantial attention in

the information extraction community, given its recent inclusion as a specific task in the NIST-sponsored Automated Content Extraction (ACE) evaluations (i.e., the ACE-2008 cross-document co-reference resolution task) or in the Text Analysis Conference (i.e. the Knowledge Base Population task, here referred to as TAC-KBP). Consider, for instance, the following sentences, each belonging to a different textual document, and consider that the word *Armstrong* is being recognized 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). Although this example considered entities sharing the same name, entities that are misspelled or that can be referenced by multiple equivalent names (e.g., *New York City*, *NYC* and *Big Apple*) should also be assigned to the same knowledge base entry.

Possible applications for named entity disambiguation include (a) enriching documents with links to authoritative Web pages on the referenced entities, (b) grouping Web search results for queries corresponding to ambiguous entities, based on their possible referents, and (c) supporting advanced information retrieval applications such as question answering and named entity search.

This work presents a thorough study on the subject of entity disambiguation, evaluating the application of a state-of-the-art learning-based approach over texts written in Portuguese or in Spanish, and considering knowledge bases consisting of articles from the Portuguese and Spanish versions of Wikipedia. It provides a discussion on the key issues involved in entity linking, as well as an empirical analysis of the effectiveness of different learning models and different sets of features. Machine learning methods for addressing ranking tasks are usually known as Learning to Rank

(L2R) approaches [19], and they have been successfully applied in document retrieval systems. However, their application in named entity disambiguation has received less attention, particularly if we consider the case of state-of-the-art L2R approaches based on ensembles of trees [12, 21, 30]. Moreover, and despite the recent interest in the entity linking task, this is to the best of my knowledge the first study that specifically addresses entity linking over Portuguese and Spanish texts.

The rest of this paper is organized as follows: Section 2 describes some of the key issues involved in the task, also presenting the usual architecture of entity linking systems. Section 3 presents related work, covering two main types of approaches, namely supervised approaches and unsupervised approaches. Section 4 presents the details of the machine learning approach, with an emphasis on the features that were used to model the relationship between named entities and candidate disambiguations. Section 5 presents the experimental results of a study comparing different configurations for the proposed machine learning approach, with different datasets, mainly focusing on the Portuguese and Spanish languages. Finally, Section 6 presents the main conclusions and points directions for future work.

2. THE ENTITY LINKING TASK

Previous works on entity linking have distinguished between two types of approaches, namely *corpus-based* (i.e., with no a priori knowledge on the entities, using clustering to disambiguate the references) and *knowledge-based* (i.e., using a knowledge base with information about each entity, usually Wikipedia, and assigning references to the most similar knowledge base entry). However, competitions designed to evaluate named entity disambiguation systems (e.g., TAC-KBP [15]) are now requiring that references should not only be assigned to a predefined knowledge base entry, but also for references to entities that are not in the knowledge base to be grouped together if they refer to the same concept, thus mixing these two types of approaches.

The general architecture for named entity linking systems, with basis on an analysis of several approaches found in the related literature, is presented in Figure 1. This general architecture consists of the following five main modules:

1. **Query expansion:** Knowledge base referents might be referenced in the texts by several alternative names, some of which might be more ambiguous. Therefore, given a reference, most systems apply expansion techniques that try to identify other names used in the source document that reference the same entity.
2. **Candidate generation:** This module filters the knowledge base entries that might correspond to the query, based on string similarity. Since Wikipedia is the most commonly used knowledge base, some of its hyperlink structure is also widely used to obtain alternative names (e.g., disambiguation pages, redirect pages, or hypertext anchors).
3. **Candidate ranking:** This module sorts the retrieved candidates according to the likelihood of being the correct referent. Most approaches use either a *Learning*

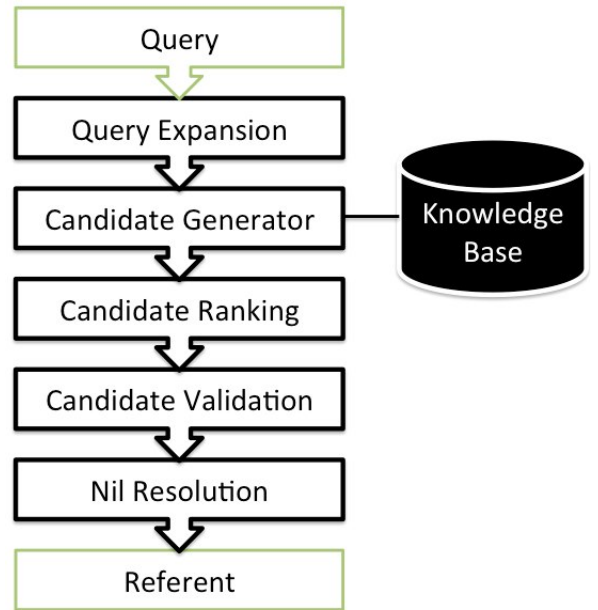


Figure 1: The general architecture of a complete named entity disambiguation system.

to Rank algorithm [19] or an heuristic method for measuring similarity between references and candidate disambiguations, such as the *Vector Space IR Model* [2].

4. **Candidate validation:** This module decides whether the top ranked referent is an error resulting from the fact that the correct referent is not present in the knowledge base. Commonly used approaches include setting a threshold on the ranking score, training a specific classification model, or applying a voting scheme.
5. **NIL entity resolution** If the system decides that a given query has no corresponding match in the knowledge base (i.e., for the case of *NIL queries*), the NIL resolution module should output an identifier specifically generated for all references to that particular entity. NIL resolution is a less studied problem and, in the context of the TAC-KBP joint evaluation, NIL resolution was only introduced in the 2011 edition.

Different types of entity linking systems, following two types of approaches, namely supervised approaches and unsupervised approaches, have been described in the literature. Moreover, we can categorize previous supervised approaches as being either candidate ranking approaches, document-level approaches or cross-document approaches. Candidate Ranking approaches model the entity linking task as a traditional ranking problem, where each mention is assigned to its most relevant candidate disambiguation [6]. Document-level approaches try to find the set of most topically coherent entities for each mention in a document, processing all entities in a same document simultaneously in an attempt to maximize both the individual relevance of each candidate and their combined coherence [7]. Finally, cross-document approaches perform the disambiguation by clustering together

documents that relate to the same entity [22].

This work addresses the entity linking problem as a candidate ranking problem, but considers a group of features computed from document-level knowledge. The following section details some of the recent proposals in the entity linking literature.

3. RELATED WORK

This section presents the most important related works concerning the entity linking task, following either (a) unsupervised approaches, or (b) supervised approaches.

3.1 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 [13]. 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 reference. With their first measure, which they called out-degree, a graph is built using the names from all document references and the textual descriptions from all candidates. Each time a reference name appears in the a candidate’s description, an edge is assigned, connecting the name and the candidate. In the end, the candidate for a reference with the most links (i.e., with the highest out-degree) is considered the correct disambiguation. The other measure considered (i.e., in-degree measure) performs the apposite approach, i.e., it searches for candidate names in the document where the reference appears.

A different unsupervised proposal by Sarmiento 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 [26]. Their approach relies on each mention being represented by a feature vector corresponding to the TF-IDF scores [20] 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.

3.2 Supervised Approaches

In this paper supervised 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 the individual occurrences.

3.2.1 Candidate Ranking Approaches

Zhang et al. [32] proposed an entity linking framework that follows two main steps, namely name variant resolution and name disambiguation. Name variant resolution 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. These variations are then used to retrieve possible candidates, through the usage of string matching. As for the name disambiguation step, a learning to rank algorithm (i.e., Ranking SVM [16]) 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 [32]. 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 iterative instance selection strategy of auto-generated training data, in order to improve the distribution of the examples generated.

Anastácio et al. presented another state-of-the art system, specifically developed to participate in the TAC-KBP 2011 challenge, that used a candidate ranking approach [1]. 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). 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). The entity linking system presented in this paper was developed as an extension of the system proposed by Anastácio et al.

3.2.2 Document-Level Approaches

The work by Cucerzan presents a named entity disambiguation approach which also relies on Wikipedia as an external knowledge repository [7]. Contrary to most other approaches, Cucerzan considers, as context for each entity reference, not the surrounding words, but instead the remaining entity references made in the same document. The proposed approach uses the traditional Vector Space IR Model, comparing document vectors with the referent vectors. The

document vector contains the categories of all possible referents for all entity references found in its text, as well as the number of occurrences of each reference. The referents have binary feature vectors with all the categories and entity references found in its Wikipedia entry. Interestingly, the similarity measure used by the author does not normalize the feature values, thus privileging important entities, which tend to have longer descriptions, more mentions, and more categories. Also, the author argues that the errors originating from the usage of the *one sense per discourse* principle [11], which simplifies the disambiguation problem through the assumption that a given document does not contain homonym entities, are non-negligible. His approach to address this problem involved determining a reference’s context in an iterative fashion. Whenever more than one Wikipedia entity scored higher than a predefined threshold, the considered context would be shrunk to the level of a paragraph, and possibly to the level of a sentence.

Ratinov et al. recently developed an approach to participate in the TAC-KBP 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 [24]. GLOW is a system previously developed by the same authors [25], 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 Mention Identification, Disambiguation and Output Reconciliation. Mention identification has the objective of identifying all mentions that might refer to the given query entity, and to assign them the most linkable Wikipedia page according to the similarity with the names of all titles, redirects and hyperlink anchors (because GLOW only links expressions that appear as hyperlinks in Wikipedia). As for the disambiguation step, it consists in the direct application of GLOW to the previous identifications. 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. Finally, the output reconciliation step decides if it should be assigned any NIL to the returned disambiguations from GLOW. This is achieved through the analysis of the scores returned from the previous step, namely the ranker score and the linker score. The first corresponds to the ranking confidence of a determined page being the correct disambiguation, whereas the linker score gives information about if it is preferable to assign a NIL instead of the disambiguation attributed.

3.2.3 Cross-Document Approaches

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 [22]. 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.

4. SUPERVISED LEARNING METHODS FOR NAMED ENTITY DISAMBIGUATION

In order to study the main aspects influencing the performance of entity linking systems in both Portuguese and Spanish texts, a prototype system was assembled. It includes the first four modules from the general architecture shown in Figure 1 (i.e., all modules except NIL resolution), and uses articles from the most recent dumps of the Portuguese and Spanish versions of Wikipedia as the knowledge bases supporting entity linking. Moreover, these knowledge bases were filtered in order to include only entries that can be considered as entities. This was achieved through the usage of the structured information made available by DBpedia¹. The complete entity linking system had already been used to participate in the TAC-KBP English entity linking task.

4.1 Overview on the Approach

I now present an overview on the proposed entity linking approach, afterwards detailing the considered features and the learning approaches for candidate ranking and validation.

In what concerns query expansion, two simple mechanisms were considered, namely one that finds acronyms for the named entity references by looking for a textual pattern that corresponds to having a set of capital words followed by the acronym inside parentheses (i.e., finding expressions like *General Electric (GE)*), or vice-versa, and another that looks for longer entity mentions in the source text (i.e., *SMART Communications* is an expansion for the query *SMART*).

As for the candidate generation module, I considered an approach that returns the top- k most likely entries in the knowledge base, according to a modified version of the traditional cosine similarity computed between the query and all knowledge base entries. The modification essentially replaces the level of the textual units being compared, making the metric work with name n -grams instead of document words, with n between 1 and 4. Roughly, this means that the more n -grams the query string has in common with a name in the knowledge base, the more probable the respective entry is to being selected as a disambiguation candidate. In this candidate generation step, I specifically set the parameter k to the value of 50, i.e., the top 50 most similar candidate names with the query name and possible name expansions are returned. However, in order to try to avoid more candidate misses, I explored the idea of retrieving possible candidates whose textual contents have a high Jaccard similarity towards the query document. This retrieval approach is achieved through the usage of a locality sensitive hashing (LSH) technique with the support of the min-hash procedure introduced by Broder [5], implemented with the

¹<http://dbpedia.org/>

objective of efficiently compute the Jaccard similarity between textual documents. Moreover, I also tried to filter the set of candidates using this approximation of the Jaccard similarity, removing the ones whose text had nothing to do with the support document. Finally, the last resource used in the candidate generation module was a dataset provided by Google containing the links to all hypertext anchors pointing to a specific Wikipedia page [27].

The candidate ranking module is based on supervised learning to rank approaches, given the success of previous works like those of Zheng et. al [33] and He et. al [14], with support of a rich set of features. The highest ranked candidates are then filtered through a supervised classifier that detects the NIL references.

4.2 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.

4.2.1 Popularity Features

It was considered a set of features that benefit more popular candidates, given the intuition that these candidates tend to be referenced more often in the texts.

- **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 the links correspond to the occurrence of hypertext links connecting the knowledge base entries. These PageRank scores capture the intuition that candidates who are linked from many other highly-linked (i.e., important) knowledge base entries should considered be more important.
- **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. This value is used as a bias, and I assume that candidates have description lengths proportional to their 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, 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.

4.2.2 Text-based Similarity

These features measure the similarity between the context where the entity reference occurs and the textual description for the candidate disambiguations.

- **Cosine Document Similarity.** The cosine similarity, using TF-IDF weights, between the candidate's description and the query source text, i.e., the document where the query occurs.
- **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.
- **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 description and the query.
- **Cosine Document Rank.** The rank of the given candidate in the list of all candidates, when candidates are ordered according to the 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.

4.2.3 Topical Similarity

This set of features leverages on topic-based representations for the query's source text and the candidate's description, as obtained through a Latent Dirichlet Allocation (LDA) topic model [3] built with basis on all textual 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. Word terms were also reduced to their corresponding stems, through stemming algorithms specific to the Portuguese, Spanish and English languages. 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 JGibbsLDA² software framework. The value of K (i.e., the considered number of topics) was adjusted empirically to 400, by looking for the number of topics that minimizes the perplexity of the LDA models in held-out Wikipedia samples.

The considered set of LDA-based topical similarity features is as follows:

- **Topic Vector Similarity.** The cosine similarity, computed between the vectors corresponding to the candidate and the query's topic probabilities.
- **Topic Match.** This feature takes the value of one if the topic that best characterizes the candidate's description is the same that best characterizes the source text, and zero otherwise.

²<http://jgibblda.sourceforge.net/>

- **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.

4.2.4 Name Similarity

These features capture similarities between the strings that make up the entity references and the corresponding candidate names. Moreover, the alternative names of each candidate are also used in the computation of these features.

- **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 candidates might have alternative names associated, the combination with the higher similarity is used as the final value.
- **Jaro-Winkler Similarity.** The highest string similarity based on the Jaro-Winkler metric between the set of all possible candidate names, and the query and respective expansions.
- **Jaccard Similarity.** The highest string similarity based on the Jaccard metric between the set of all possible candidate names, and the query and respective expansions.

- **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.

4.2.5 Entity features

These features leverage on results from Named Entity Recognition (NER) systems, capable of processing texts written in Portuguese, Spanish or english, applied to both the query’s source text and the candidate’s description. NER systems return not only the named entities occurring in the text but also their estimated type (i.e., person, organization, or location). I specifically used a learning-based method for Named Entity Recognition, relying on the implementation provided by the Stanford NER³ package.

For the Portuguese language, a CRF-based NER model was trained using the written portion of the CINTIL International Corpus of Portuguese⁴, which is composed of 726.916 word tokens taken from texts collected from different sources and domains. For the Spanish language, we used a similar model trained with the Spanish dataset of the CoNLL 2002 Shared Task on Named Entity Recognition⁵, which contains 380.926 word tokens taken from newswire articles made available by the Spanish EFE News Agency. In both cases, the CRF models used a standard set of features that included token identity within a window of 3 tokens, token prefixes and suffixes, morpho-syntactic categories, and token type patterns (e.g., uppercase versus mixed case or numeric versus alphanumeric). When using a ten-fold cross-validation experiment, the Portuguese NER model achieved an average *F1* score of 76.0% whereas the Spanish NER model achieved an *F1* score of 79%. For processing English texts, I used the default model distributed with the Stanford NER package.

The considered set of entity 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 present in 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. Each type is represented by a binary feature.

³<http://nlp.stanford.edu/software/CRF-NER.shtml>

⁴http://catalog.elra.info/product_info.php?products_id=1102

⁵<http://www.lsi.upc.edu/~nlp/tools/nerc/nerc.html>

- **Candidate Type:** Similar to the query type, but based on the information available in the knowledge bases, as both 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.

4.2.6 Geographic Features

These 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 [17] or of Lieberman and Samet [18], 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.
- **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.
- **Missed Geo Entities.** The number of place references in the source text that are not mentioned in the candidate's textual description.
- **Geospatial Distance.** It is used an efficient similarity search method based on min-hash [5] 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 Vicenty [28].
- **Geospatial Containment.** It is 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 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 JTS⁶ software framework.

An extensive evaluation on the impact of geographic features is presented in my MSc thesis report. Due to space restrictions, the results are not shown in this paper but, in the tests performed, the results showed only marginal improvements when introducing this new set of features in the training of entity linking models.

4.2.7 Document-Level Features

Many previous works addressing the entity linking task have noted that significant improvements could be achieved by performing entity disambiguation at a document level [7], 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 the feature that we called *contextual PageRank*.

⁶<http://www.vividsolutions.com/jts/JTSHome.htm>

The software used to calculate the PageRank scores was the WebGraph⁷ package from the University of Milan.

4.2.8 Validation-Only Features

Besides the aforementioned features, and noticing that the validation module only takes as input the top ranked candidate (i.e., the best possible disambiguation), we considered some 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. These features are only considered for the case of the validation model, and their values are computed after ranking has already been made.

- **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 to the query's set of candidate.
- **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 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 second-ranked candidate, 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.

4.3 Supervised Learning Models

During this work, I experimented with two different Learning to Rank (L2R) methods for building the candidate ranking model, namely with (i) a pairwise Ranking SVM algorithm [16], and with (ii) an ensemble of LambdaMART [12] models obtained through the Random Forests technique [4].

Ranking SVM, which is a commonly used L2R method, transforms the ranking problem into a set of binary classification tasks which are addressed through the formalism of Support Vector Machines (SVM). LambdaMART is, on the other hand, a state-of-the-art boosting technique for ranking problems, which combines the flexibility of boosted least squares regression trees, through the formalism of MART, with the idea of using functional gradients computed through the product of a pairwise cross-entropy loss, applied to the logistic of the model scores, with the listwise delta in the considered evaluation metric that is gained by swapping the pair of items. Both these ranking methods are available through

⁷<http://webgraph.di.unimi.it/>

open-source packages, namely through the SVMrank⁸ and RankLib⁹ libraries.

As for the candidate validation module, I experimented with an SVM classifier with a Radial Basis Function as kernel, and also with a Random Forest classifier [4]. These classification algorithms are available through the SVMlight¹⁰ and Weka¹¹ open-source software libraries.

Each of the considered techniques models the problem with different levels of complexity. For instance Random Forest classifiers combine several tree models, which try to define a function that logically partitions the classification space in terms of a tree of decisions made over attributes of the original data. SVMs, on the other hand, use a kernel function to flexibly map the original data into a higher-dimensional space, where a separating hyper-plane can be defined. It has been shown that SVMs exhibit a high resistance to noise, handle correlated features well, and rely only on the most informative training examples, which leads to a larger independence from the relative sizes of the sets of positive and negative examples. They are currently the most widely-used classification technique. Random Forests, on the other hand, are a more recently proposed ensemble method, consisting of many decision trees and combining bagging with the random selection of features. This is currently one of the best learning algorithms available, although Random Forests do not handle large numbers of irrelevant features as well as other learning methods.

5. EXPERIMENTAL VALIDATION

I compared 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., through the automatic generation of training and test data) 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-21¹² dataset. Moreover, in order to compare the results obtained over these two languages against 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

⁸http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

⁹<http://www.cs.umass.edu/~vdang/ranklib.html>

¹⁰<http://svmlight.joachims.org/>

¹¹<http://weka.wikispaces.com/>

¹²<http://hltcoe.jhu.edu/datasets/>

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 1: Number of entries considered in each knowledge base.

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 [15], in tests with documents in other languages.

Table 2 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 belonging to one of three groups, namely people, organizations, and geopolitical entities. As for the Knowledge Bases (KBs) 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 2.

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.

5.1 Learning Algorithms

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

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 than others [31], 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 3, 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, 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, showing that the disambiguation

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%
LambdaMART	KBP-13 ES	68.6%	68.1%	74.1%
	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 3: Results for the different L2R algorithms.

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

Table 4: Detailed results for the best configuration.

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 5 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 82.2%) participating in TAC-KBP 2009 was based on a TF/IDF ranking model.

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 5: Results for the Random Forest models trained with the English TAC-KBP data.

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 [8], 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.

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 from previous editions can not be made. Although, we can see that the system achieves close in performance comparing to the previous years best systems (i.e., when using the SVMRank as ranking algorithm). 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.

5.2 Candidate Selection

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 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 6. 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*).

Dataset	Candidate Misses	% of total queries	Overall Accuracy
XLEL-21 PT	01	0.584%	97.5%
XLEL-21 ES	01	0.735%	94.7%
Wiki PT	03	0.025%	97.9%
Wiki ES	06	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 6: Number of candidate misses.

Table 7 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 from 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 [?]. 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, 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 than 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.

5.3 Feature Contribution

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.

Ranking accuracy results, for the Portuguese and Spanish Wikipedia datasets, are presented in Table 8, using our best performing configuration (i.e., LambdaMART models for ranking, and Random Forest models for validation).

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 8: Accuracy after removing sets of features.

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 9 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 different values for

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 7: Results for different configurations of the candidate retrieval module.

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

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.

6. CONCLUSIONS AND FUTURE WORK

Despite the recent advances in the named 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 this work, I specifically focused on the application to the Spanish and Portuguese languages. 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 meth-

ods to achieve a performance in line with that of current state-of-the-art approaches, as reported on experiments focusing on the English language.

For future work, it would be interesting to experiment with the usage of additional features. It is possible that features derived from structured information associated to the knowledge based entries (i.e., features derived from slot-filling methods) could provide 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 [23]. 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 paper only addressed the disambiguation of named entity references, assuming the existence of a named entity recognition system. In the case of this experiments, the entities to be disambiguated were explicitly provided as queries, and I 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 named entity recognition approach that was proposed by [9].

7. REFERENCES

- [1] Ivo Anastácio, Pável Calado, and Bruno Martins. Supervised learning for linking named entities to wikipedia pages. In *Proceedings of the Text Analysis Conference*, 2011.
- [2] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [3] David M. Blei, Andrew Y. Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.
- [4] Leo Breiman. Random forests. *Machine Learning*, 45(1), 2001.
- [5] Andrei Z. Broder. On the resemblance and containment of documents. In *Proceedings of the IEEE Conference on Compression and Complexity of Sequences*, 1997.
- [6] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the European Conference of the Association for Computational Linguistic*, EACL '06, 2006.
- [7] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, pages 708–716, 2007.
- [8] Silviu Cucerzan. Tac entity linking by performing full-document entity extraction and disambiguation. In *Proceedings of the Text Analysis Conference*, 2011.
- [9] Doug Downey, Matthew Broadhead, and Oren Etzioni. Locating complex named entities in web text. In *Proceedings of the 20th international joint conference on Artificial intelligence*, IJCAI'07, pages 2733–2739. Morgan Kaufmann Publishers Inc., 2007.
- [10] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the web. *Commun. ACM*, 51:68–74, 2008.
- [11] William A. Gale, Kenneth W. Church, and David Yarowsky. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, HLT '91, pages 233–237. Association for Computational Linguistics, 1992.
- [12] Yasser Ganjisaffar, Rich Caruana, and Cristina Videira Lopes. Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 85–94, New York, NY, USA, 2011. ACM.
- [13] Yuhang Guo, Wanxiang Che, Ting Liu, and Sheng Li. A graph-based method for entity linking. In *Proceedings of the International Joint Conference on Natural Language Processing*, 2011.
- [14] Jiyin He and Maarten de Rijke. A ranking approach to target detection for automatic link generation. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 831–832. ACM, 2010.
- [15] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.
- [16] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142. ACM, 2002.
- [17] JL Leidner. *Toponym Resolution: a Comparison and Taxonomy of Heuristics and Methods*. PhD thesis, PhD Thesis, University of Edinburgh, 2007.
- [18] Michael Lieberman and Hanan Samet. Adaptive context features for toponym resolution in streaming news. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012.
- [19] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.
- [20] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [21] Ananth Mohan, Zheng Chen, and Kilian Q. Weinberger. Web-search ranking with initialized gradient boosted regression trees. *Journal of Machine Learning Research, Workshop and Conference Proceedings*, 14:77–89, 2011.
- [22] S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung. Cross-lingual cross-document coreference with entity linking. In *Proceedings of the Text Analysis Conference*, 2011.
- [23] Tao Qin, Tie-Yan Liu, Xu-Dong Zhang, De-Sheng Wang, Wen-Ying Xiong, and Hang Li. Learning to rank relational objects and its application to web search. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 407–416. ACM, 2008.
- [24] Lev Ratnov and Dan Roth. Glow tac-kbp 2011 entity linking system. In *Proceedings of the Text Analysis Conference*, 2011.
- [25] Lev Ratnov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.
- [26] L. Sarmento, A. Kehlenbeck, E. Oliveira, and L. Ungar. An Approach to Web-Scale Named-Entity Disambiguation. In *Proceedings of the 6th International Conference on Machine Learning and Data Mining in Pattern Recognition*, MLDM '09, pages 689–703. Springer-Verlag, 2009.
- [27] Valentin I. Spitzkovsky and Angel X. Chang. A cross-lingual dictionary for english wikipedia concepts. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, 2012.
- [28] Thaddeus Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review XXIII*, 176, 1975.
- [29] Casey Whitelaw, Alex Kehlenbeck, Nemanja Petrovic, and Lyle Ungar. Web-scale named entity recognition. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 123–132. ACM, 2008.
- [30] Q. Wu, C. J. C. Burges, K. Svore, and J. Gao. Adapting boosting for information retrieval measures.

Journal of Information Retrieval, 2007.

- [31] Emine Yilmaz and Stephen Robertson. On the choice of effectiveness measures for learning to rank. *Information Retrieval*, 13(3), 2010.
- [32] Wei Zhang, Jian Su, and Chew-Lim Tan. A wikipedia-lda model for entity linking with batch size changing instance selection. In *Proceedings of International Joint Conference on Natural Language Processing*, 2011.
- [33] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 483–491. Association for Computational Linguistics, 2010.