Journal of Artificial Intelligence Research 49 (2014) 733-773 Submitted 07/13; published 04/14

Comparative Evaluation of Link-Based Approaches for Candidate Ranking in Link-to-Wikipedia Systems

Norberto Fernández García berto@it.uc3m.es
Jesús Arias Fisteus jaf@it.uc3m.es
Luis Sánchez Fernández luiss@it.uc3m.es
Telematics Engineering Department
Universidad Carlos III de Madrid
Avda. Universidad, 30, E-28911
Leganés, Madrid, Spain.

Abstract

In recent years, the task of automatically linking pieces of text (anchors) mentioned in a document to Wikipedia articles that represent the meaning of these anchors has received extensive research attention. Typically, link-to-Wikipedia systems try to find a set of Wikipedia articles that are candidates to represent the meaning of the anchor and, later, rank these candidates to select the most appropriate one. In this ranking process the systems rely on context information obtained from the document where the anchor is mentioned and/or from Wikipedia. In this paper we center our attention in the use of Wikipedia links as context information. In particular, we offer a review of several candidate ranking approaches in the state-of-the-art that rely on Wikipedia link information. In addition, we provide a comparative empirical evaluation of the different approaches on five different corpora: the TAC 2010 corpus and four corpora built from actual Wikipedia articles and news items.

1. Introduction

Due to the important volume of information contained in Wikipedia, but also to the open nature of this content, the on-line encyclopedia has been adopted in recent times as a useful resource for computational linguistics tasks like name translation (Lin, Snover, & Ji, 2011), named entity recognition (Nothman, Murphy, & Curran, 2009), etc.

The development of automatic link discovery systems (Erbs, Zesch, & Gurevych, 2011) is another area of research where Wikipedia has had an important impact. The task of discovering links to Wikipedia articles has been addressed, with slight variants and under different names, by different communities. For instance, Hachey et al. (2013) distinguish between *named entity linking*, addressed in the context of the Knowledge Base Population (KBP) track (National Institute of Standards and Technology, 2014b) at the Text Analysis Conference (TAC) (National Institute of Standards and Technology, 2014a), and *wikification*, addressed in the Link-the-Wiki track at the Initiative for the Evaluation of XML retrieval (INEX) (INEX, 2014). In both cases the goal is to automatically find Wikipedia articles that represent the meaning of a certain piece of text in the document and define a link to Wikipedia using as anchor that piece of text. However, there are differences in aspects like the anchors considered (only named entities in *named entity linking*, named entities and common terms in *wikification*) or in whether the Wikipedia is considered as a complete source of knowledge (*wikification*) or not (*named entity linking*).

Henceforth, we will simply refer as *link-to-Wikipedia* to the general task of discovering links to Wikipedia, which includes both *wikification* and *named entity linking* as particular cases.

According to Erbs et al. (2011), the task of discovering links can be divided into a series of steps. They include: identifying the anchors to be linked, searching for candidate link targets for each anchor, and selecting the best candidate from the results of the searching step. It is common that link-to-Wikipedia approaches address these steps independently and sequentially (though there are also examples where the steps are not independent, like Cucerzan, 2012 or Sil, 2013).

Due to its important role (Ji, Grishman, & Dang, 2011), in the context of this paper we will center our attention in the last of the aforementioned processes. It is referred to as *disambiguation* by Hachey et al. (2011). However, as selecting the best link target usually involves creating a ranking of all the candidates to choose the one with the highest rank, other authors refer to the process as *target ranking* (Erbs et al., 2011) or *candidate ranking* (Guo, Tang, Che, Liu, & Li, 2011; Ploch, Hennig, de Luca, & Albayrak, 2011; Ji et al., 2011). In this article, we will also adopt the term *candidate ranking*.

In order to select the best Wikipedia article to link from a given anchor, the candidate ranking process relies on the context information provided by a set of features. These features are extracted from the document where the anchor is placed (the *context document*) and/or the different Wikipedia articles considered as candidates to become the link target. According to Erbs et al. (2011) the features can be classified into three groups: (1) those extracted from the text of the document/articles, (2) those extracted from their titles; and (3) those based on existing links. The latter are the subject of study in this paper.

Traditional research in the area of computational linguistics has shown the effectiveness of using WordNet (Miller, 1995) graph links for tasks like computing semantic relatedness (Budanitsky & Hirst, 2006) or performing word sense disambiguation (Navigli & Lapata, 2010). In the case of link discovery, Erbs et al. (2011) indicate that, when enough training information is available, link-based approaches can outperform text-based ones. Taking this into account, it is not surprising to find many link-to-Wikipedia approaches that use features for candidate ranking based only on information about links. Some examples are the work of Milne and Witten (2008b), Pilz (2010), Radford et al. (2010), Fernández et al. (2010), Guo et al. (2011), Ploch et al. (2011) and Ratinov et al. (2011).

Given the ample variety of link-based features for candidate ranking described in the state of the art, a comparative analysis of the different alternatives can be useful to decide which approach (or approaches) should be considered when designing link-to-Wikipedia systems. However, using only the results published in the state of the art it is difficult to compare across systems the ranking performance of the different link-based approaches. First, link-to-Wikipedia systems are usually evaluated in an end-to-end setup, that is, the evaluation involves not only the ranking stage, but also the candidate searching and the candidate selection processes. Thus, the impact in performance of the different system components is mixed. Second, in general, link-to-Wikipedia systems do not only rely on link-based features to rank the candidates, but also combine them with features of the other types. Thus, the effects of the different contributions to the ranking process are also mixed.

Taking this into account, the main goals of this paper are twofold: (1) offer an overview of link-based approaches for candidate ranking in link-to-Wikipedia systems; and, (2) perform an empirical evaluation to compare these approaches. In order to address the aforementioned difficulties, we: (1) focus our analysis on the candidate ranking stage, isolating it as much as possible from the candidate search/selection stages; and (2) consider only link-based features, not combined with those based on text or titles. A similar comparison is, to the knowledge of the authors, not available at the time of writing.

The rest of this paper is organized as follows: section 2 presents some definitions and a formal description of the problem to be addressed. Section 3 outlines the different link-based approaches to be compared. Section 4 describes the setup of the empirical evaluation we have carried out, as well as its results. Section 5 offers an overview of related work. Finally, section 6 closes this paper with concluding remarks and future lines of work.

2. Definitions and Problem Formalization

In this section we introduce some definitions and nomenclature that will be helpful in the rest of the article.

The textual document that mentions the anchor a that is going to be linked to Wikipedia is named *context document* and will be represented by d_c.

The set of all the *Wikipedia articles* will be denoted by W, whereas particular Wikipedia articles will be represented by w_i, $i = 1, \dots, |W|$. In our case we will consider in the set W only the Wikipedia pages that belong to the *Main* namespace (Wikipedia, 2014b) and that represent non-ambiguous concepts (that is, disambiguation pages will be filtered out).

We will denote as $C(d_c, a)$ the set of Wikipedia articles $\{c_1, c_2, \dots, c_{|C(d_c,a)|}\}$, $c_k \in W$ that are selected as *candidates* to fit the meaning of a in d_c.

A *link* l can be defined as a duple $l = (src(l), dest(l))$, where $src(l)$ represents the document that is the *source* of the link and $dest(l)$ is the document that is pointed by the link, that is, its *destination*.

We will denote as $F(d)$ the set of documents that are destination of the *forward links* from d, that is: $F(d) = \{f \mid \exists l,\, src(l) = d \wedge dest(l) = f\}$. Similarly, we will represent as $B(d)$ the set of documents that are the source of the *backward links* of d, that is: $B(d) = \{b \mid \exists l,\, src(l) = b \wedge dest(l) = d\}$. In this paper, we use only the information provided by Wikipedia links. Thus, we will only consider Wikipedia articles as members of $F(d)$ and $B(d)$. Note also that both $F(d)$ and $B(d)$ are sets and, thus, they do not consider duplicates. However, it might happen that a document has several links pointing to the same destination. In order to represent this information, we will denote the number of links that have as source the document s and as destination the document d as $n(s, d)$.

Taking into account the aforementioned definitions, the candidate ranking process to be addressed in the context of this paper may be formalized as follows:

Definition 1. *Given a context document d_c, which mentions an anchor a, and a set of candidate Wikipedia articles $C(d_c, a)$, the* candidate ranking *task consists on ordering the members of $C(d_c, a)$ according to a rating. This rating measures the suitability of each candidate to represent the meaning of the anchor. The candidate $c_i \in C(d_c, a)$ with the highest rank, which fits best with the meaning of the anchor a in the context of the document d_c, is then selected to define a new link (d_c, c_i).*

A few aspects should be stressed from this definition:

- We do not introduce any restriction on the nature of the anchors to be linked. In particular, they may represent either named entities (persons, organizations, etc.) or common terms.
- As in some previous related work (Cucerzan, 2007; Mihalcea & Csomai, 2007; Han, Sun, & Zhao, 2011), we do not address in this paper the scenario where an adequate Wikipedia article to be linked to a does not exist.

3. Overview of Approaches

We describe here the different candidate ranking approaches that are evaluated in this paper. They have in common that their only source of context information are links. In particular, the links considered are those available in the context document, d_c, as well as those that constitute the link structure of Wikipedia, including the links to/from the candidate Wikipedia articles, $c_i \in C(d_c, a)$.

However, not all the approaches that are considered use the link information in the same manner. In particular, we classify them in two families, which we name (1) *bag-of-links* approaches, and (2) *graph* approaches. The main difference between them is that in the second group the links are used to build a graph structure, which is later analyzed to select the best candidate. This is not the case of the approaches in the first group.

3.1 Bag-of-Links Approaches

In this section we present a set of approaches with a characteristic in common: they do not rely on building a graph with the link context information to rank the candidates.

Nevertheless, the approaches in this family have also differences in the way they address the task. In particular, we can distinguish at least three alternative groups:

- Some approaches rely on *similarity metrics* that compute a similarity score between the context document and each candidate, to later select the candidate with the highest score (the most similar one).
- Another alternative is to rely on *popularity metrics*, which simply try to compute a popularity score for each candidate. The most popular candidate is selected. These approaches do not rely on the information provided by the context document.
- The ranking process can also be modeled as a *statistical* problem. Statistical methods are used to select the most likely candidate, given the context information.

In accordance with this classification, the following sections describe each group of approaches.

3.1.1 Similarity Metrics

The first similarity metrics we consider is the *relatedness*, computed on the basis of the Wikipedia Link-based Measure (Milne & Witten, 2008a). The relatedness is used as feature in link-to-Wikipedia approaches such as that of Milne and Witten (2008b), Han and Zhao (2009), Kulkarni et al. (2009), Pilz (2010), Fahrni et al. (2011), Han et al. (2011) and Ratinov et al. (2011).

Basically, the relatedness allows to compute the similarity between two Wikipedia documents w_i, w_j from the links they have in common. In the original definition by Milne and Witten (2008a), it can be computed as:

$$R_B(w_i, w_j) = \frac{log(max\{|B(w_i)|, |B(w_j)|\}) - log(|B(w_i) \cap B(w_j)|)}{log(|W|) - log(min\{|B(w_i)|, |B(w_j)|\})} \tag{1}$$

According to Milne and Witten (2008a), the relatedness metrics is based on the Normalized Google Distance (NGD), defined by Cilibrasi and Vitanyi (2007). The NGD is based on the intuition that terms that have a similar or related meaning co-occur frequently in documents. Thus, given a pair of terms, the Google search engine can be used to obtain pages which mention these terms. Pages that mention both of them indicate relatedness, while pages with only one of them suggest unrelatedness. As indicated by Milne and Witten (2008a), the relatedness metrics, as defined in equation 1, shares the same inspiring principle, but uses Wikipedia links instead of Google search results to account for mentions.

Being a distance metrics, relatedness values are expected to be smaller the more similar the Wikipedia articles are. However, it is easy to transform the distance metrics into a similarity metrics following the approach of Gracia and Mena (2008), which requires the computation of:

$$sim_{R_B}(w_i, w_j) = e^{-2R_B(w_i, w_j)} \tag{2}$$

A second similarity metrics between Wikipedia articles to be considered is based on computing the *Pointwise Mutual Information* (PMI) between the sets of links in the articles to be compared. For instance, it is used by Ratinov et al. (2011) for the the link-to-Wikipedia task. It is defined in that work as:

$$P_B(w_i, w_j) = \frac{|B(w_i) \cap B(w_j)|/|W|}{(|B(w_i)|/|W|)(|B(w_j)|/|W|)} \tag{3}$$

Note that the definitions of equations (1) and (3) rely on backlinks ($B(x)$) for computation. However, as indicated by Ratinov et al. (2011), both the relatedness and the PMI can also be computed using the outgoing links from a document. In this paper we explore and compare both alternatives and denote the relatedness similarity and the PMI computed with forward links as sim_{R_F} and P_F respectively.

Taking the aforementioned definitions into account, for each Wikipedia article $\{c_1, \ldots, c_k\} \in C(d_c, a)$ we can compute its relatedness and PMI with each of the Wikipedia articles linked from d_c, that is, with the $f_j \in F(d_c)$. Combining these different values we obtain the final relatedness and PMI between c_i and d_c. According to Ratinov et al. (2011) several ways to combine the values may be followed, such as taking their average or the maximum. We explore these different possibilities in the paper. In particular, for the relatedness:

$$Rel_F^A(c_i, d_c) = \frac{1}{|F(d_c)|} \sum_{\forall f_j \in F(d_c)} sim_{R_F}(c_i, f_j) \tag{4}$$

$$Rel_B^A(c_i, d_c) = \frac{1}{|F(d_c)|} \sum_{\forall f_j \in F(d_c)} sim_{R_B}(c_i, f_j) \tag{5}$$

$$Rel_F^M(c_i, d_c) = \max_{\forall f_j \in F(d_c)} sim_{R_F}(c_i, f_j) \tag{6}$$

$$Rel_B^M(c_i, d_c) = \max_{\forall f_j \in F(d_c)} sim_{R_B}(c_i, f_j) \tag{7}$$

Whereas the Pointwise Mutual Information can be computed as:

$$PMI_F^A(c_i, d_c) = \frac{1}{|F(d_c)|} \sum_{\forall f_j \in F(d_c)} P_F(c_i, f_j) \tag{8}$$

$$PMI_B^A(c_i, d_c) = \frac{1}{|F(d_c)|} \sum_{\forall f_j \in F(d_c)} P_B(c_i, f_j) \tag{9}$$

$$PMI_F^M(c_i, d_c) = \max_{\forall f_j \in F(d_c)} P_F(c_i, f_j) \tag{10}$$

$$PMI_B^M(c_i, d_c) = \max_{\forall f_j \in F(d_c)} P_B(c_i, f_j) \tag{11}$$

Another well-known approach to compute document similarity within the natural language processing and information retrieval communities is the *cosine similarity*. Basically, a vector is built to represent the context document and each candidate article. Then, the similarity between the context document and each candidate is computed as the cosine of the angle between the respective vectors. Several approaches in the state of the art (Bunescu & Pasca, 2006; Fader, Soderland, & Etzioni, 2009; Nguyen & Cao, 2010; Fahrni et al., 2011; Ploch et al., 2011; Ratinov et al., 2011) use the cosine similarity. However, there are differences between them in the features used to build the vector representations for the documents.

In our case, we have the requirement of considering solely links as context information. Thus, each document will be represented using only the links that are mentioned in that document. A similar approach to model documents and compute their cosine similarity is used, for instance, by Fahrni et al. (2011) and Ploch et al. (2011).

In particular, a document d is represented as a vector $v_d \in \mathbb{R}^{|W|}$. Each component $v_{d,i}$, $i = 1, \ldots, |W|$ of the vector v_d can be computed with the traditional term frequency (TF), inverse document frequency (IDF) product (Manning, Raghavan, & Schtze, 2008) as follows:

$$v_{d,i} = TF(d, w_i) \cdot IDF(w_i) = \frac{n(d, w_i)}{\sum_{\forall w_j \in F(d)} n(d, w_j)} \cdot log \frac{|W|}{|B(w_i)|} \tag{12}$$

Note that if $F(d)$ does not contain a certain Wikipedia article w_i, then $n(d, w_i) = 0$, $TF(d, w_i) = 0$ and, thus, $v_{d,i} = 0$. Due to this, the vector v_d is expected to be sparse.

Given two documents to be compared (for instance, d_c and a Wikipedia article $c_i \in C(d_c, a)$), the cosine similarity metrics is computed as the cosine of the angle between the vectors of the two documents, as follows:

$$sim_{cos}(v_{c_i}, v_{d_c}) = \frac{v_{d_c} \cdot v_{c_i}}{||v_{d_c}||_2 \; ||v_{c_i}||_2} \tag{13}$$

Finally, Radford et al. (2010) suggest a metrics based on the Wikipedia link structure, which can also be interpreted as a similarity metrics. In order to compute this metrics, the following equation is computed for each candidate $c_i \in C(d_c, a)$:

$$sim_R(c_i, d_c) = log(|B(c_i) \cap L_{d_c}| + 1) + 1 \tag{14}$$

where L_{d_c} is the set built by the union of all the backlinks of all the Wikipedia articles linked from d_c:

$$L_{d_c} = \bigcup_{\forall f_i \in F(d_c)} B(f_i) \tag{15}$$

All the aforementioned similarity metrics can be trivially used to address the candidate ranking process. The candidate $c_i \in C(d_c, a)$ to be selected as link destination has a maximal similarity with the context document d_c:

$$\arg\max_{c_i} \{simf(c_i, d_c)\} \tag{16}$$

where $simf$ represents one the functions: Rel_F^A, Rel_F^M, Rel_B^A, Rel_B^M, PMI_F^A, PMI_F^M, PMI_B^A, PMI_B^M, sim_{cos}, sim_R.

3.1.2 Popularity Metrics

Algorithms based on popularity metrics constitute the second group of the bag-of-links family.

A first approach that could be used to compute the popularity of a certain candidate, $c_i \in C(d_c, a)$, is simply counting the number of Wikipedia articles that link to it, that is, its *indegree*, $|B(c_i)|$ or, alternatively, the number of Wikipedia articles linked from it, its *outdegree*, $|F(c_i)|$. These metrics are considered, for instance, in the work of Dredze et al. (2010), Guo et al. (2011) and Cao et al. (2011).

Fader et al. (2009) describe a popularity score that is also based on the incoming links from Wikipedia to a candidate c_i:

$$pop_F(c_i) = (1 + log(1 + \frac{|B(c_i)|}{\alpha})) \tag{17}$$

where α is a parameter that is set to $\alpha = 15$ (Fader et al., 2009).

Finally, the *degree centrality* of a certain Wikipedia candidate article c_i can also be considered as a bag-of-links popularity metrics. Hachey et al. (2011) define the degree centrality as:

$$D(c_i) = \frac{|B(c_i)|}{|W| - 1} \tag{18}$$

Note that, as indicated at the beginning of this section, the aforementioned popularity metrics do not take into account the information provided by the context document. All of them depend only on information obtained from the candidates.

All the aforementioned metrics can be used to rank the candidates by popularity. Then, the most popular candidate $c_i \in C(d_c, a)$ is selected as the link destination. Taking into account that all the functions of $|B(c_i)|$ involved in the aforementioned approaches (linear, logarithm) are monotonically increasing functions, the order (ranking) provided in all the cases should be the same. Due to this, in the context of this paper, we consider for evaluation the indegree and outdegree only:

$$\arg\max_{c_i}\{\text{indegree}(c_i)\} \tag{19}$$

$$\arg\max_{c_i}\{\text{outdegree}(c_i)\} \tag{20}$$

3.1.3 Statistical Techniques

The candidate ranking process that we are addressing in the context of this paper can also be mathematically modeled using statistical techniques, as has been suggested in the work of Fader et al. (2009) and Han and Sun (2011).

In our particular case, considering the set of Wikipedia articles linked from d_c, $F(d_c) = \{f_1, \ldots, f_{|F(d_c)|}\}$ as input features, the destination for a can be computed by selecting the Wikipedia article $c_i \in C(d_c, a)$ that maximizes the conditional probability:

$$P(c_i/f_1, \ldots, f_{|F(d_c)|}) \text{ where } f_i \in F(d_c) \tag{21}$$

If the number of features $|F(d_c)|$ to be considered is relatively large, estimating the values of the conditional probability in equation (21) for each c_i would be a complex problem. Due to this, in practice, the problem is reformulated to make it more treatable. In particular: (1) the Bayes rule is used to reverse the conditional probability in equation (21); and, (2) it is assumed that the features (links in $F(d_c)$ in our case) are conditionally independent (*Naive Bayes assumption*).

The result of this problem reformulation is known in the state of the art as the *Naive Bayes classifier* (Manning et al., 2008). In our specific scenario, this classifier should be able to distinguish which of the *classes* (the different $c_i \in C(d_c, a)$) is the most likely for the anchor a.

Mathematically, the expression that we use to select the best c_i using the *maximum a posteriori* decision rule (Manning et al., 2008) is:

$$\arg\max_{c_i}\{NB(c_i, d_c)\} = \arg\max_{c_i}\{\log P(c_i) + \sum_{j=1}^{|F(d_c)|} n(d_c, f_j) \log P(f_j/c_i)\} \tag{22}$$

where the logarithm function is used to avoid underflows (as suggested in Manning et al., 2008).

In order to compute the values in equation (22) for each c_i, we need to know the value of two probabilities: (1) the *prior* probability of class c_i, $P(c_i)$; and, (2) the conditional probabilities $P(f_j/c_i)$. To estimate these two probabilities we follow the approach described by Manning et al. (2008):

$$P(c_i) = \frac{\sum_{\forall b_j \in B(c_i)} n(b_j, c_i)}{\sum_{\forall w_i \in W} \sum_{\forall f_j \in F(w_i)} n(w_i, f_j)} \tag{23}$$

That is, $P(c_i)$ represents the Maximum Likelihood Estimate (MLE) of the probability that a certain document contains a link to c_i, computed by dividing the number of actual links to c_i by the total number of links in Wikipedia:

$$P(f_j/c_i) = \frac{1 + \sum_{\forall b_i \in B(c_i)} n(b_i, f_j)}{\sum_{\forall w_j \in W}(1 + \sum_{\forall b_i \in B(c_i)} n(b_i, w_j))} \tag{24}$$

In this case, $P(f_j/c_i)$ represents the probability of having an anchor linking to f_j when the document already contains a link to c_i. Again, the MLE is also used for the conditional probabilities and, thus, the probabilities are computed by dividing the number of links to f_j in documents that contain a link to c_i by the total number of links in documents that contain a link to c_i. It can be seen that the MLE is smoothed using the *Laplace smoothing* to avoid zeros.

3.2 Graph Approaches

The second family of link-based approaches for candidate ranking we consider are the graph approaches, which rely on building a graph and processing it to select the best candidate.

3.2.1 PageRank and Personalized PageRank

The first algorithm that we consider within the graph family is PageRank, first defined by Page et al. (1999), and widely known due to its use as part of the Google search engine. Examples of application of PageRank as a method for candidate ranking can be found for instance in the work of Fernández et al. (2010), Dredze et al. (2010) and Hachey et al. (2011).

Basically, PageRank is an algorithm that can be used to compute the popularity of a certain page, taking into account the popularity and number of pages that link to it. Using the mathematical formulation described by Brin and Page (1998) in the particular scenario that we are addressing in this paper, to compute the popularity $PR(w_i)$ of a Wikipedia article w_i, we need to solve the following equation:

$$PR(w_i) = \frac{(1-d)}{|W|} + d \cdot [\sum_{\forall w_j \in B(w_i)} \frac{1}{|F(w_j)|} PR(w_j)\ ] \tag{25}$$

Where d is a damping factor which can be set between 0 and 1, but is typically set to 0.85 according to Brin and Page (1998) and Hachey et al. (2011).

Note that, according to equation (25), we are only taking into account links to/from Wikipedia, because computing PageRank in the general scenario requires complete information of the link structure of the Web, which is a computationally expensive problem. The same simplification is also assumed by Fernández et al. (2010) and Hachey et al. (2011).

Note also that, as it happens with the popularity metrics described in section 3.1.2, the PageRank metrics does not depend on the context information in d_c, but only on the graph built from the link structure of Wikipedia. However, the context information in d_c can be included in the process by using a variant of the algorithm known as *Personalized PageRank* or *Topic-Sensitive PageRank* (Haveliwala, 2003). This algorithm is used for instance by Yeh et al. (2009) to define a semantic relatedness metrics.

The main difference between classical PageRank and Personalized PageRank is that, instead of relying on a uniform damping vector, it is biased to give more relevance to a given set of resources (Haveliwala, 2003). In our particular case, these resources are the articles linked from d_c, that is, the members of $F(d_c)$. In practice, equation (25) is adapted as follows to compute the Personalized PageRank:

$$PPR(w_i, d_c) = \begin{cases} d \cdot [\sum_{\forall w_j \in B(w_i)} \frac{1}{|F(w_j)|} PPR(w_j, d_c)] & \text{if } w_i \notin F(d_c) \\ (1-d) \cdot TF(d_c, w_i) + \quad d \cdot [\sum_{\forall w_j \in B(w_i)} \frac{1}{|F(w_j)|} PPR(w_j, d_c)] & \text{if } w_i \in F(d_c) \end{cases}$$

where $TF(d_c, w_i)$ represents the term frequency of the link to w_i in the context of the document d_c, computed as indicated in equation (12).

Once the PageRank and Personalized PageRank values are computed, they can be used to rank the candidates. The article $c_i \in C(d_c, a)$ with the highest $PR(c_i)$ or $PPR(c_i, d_c)$ value is then selected as the link destination:

$$\arg\max_{c_i} \{PR(c_i)\} \tag{26}$$

$$\arg\max_{c_i} \{PPR(c_i, d_c)\} \tag{27}$$

3.2.2 Random Walk

Several works in the state of the art (Gentile et al., 2009; Fernández et al., 2010; Han et al., 2011; Ploch et al., 2011; Jiménez et al., 2013) define techniques to link several anchors in the same context document at the same time. Usually, these approaches address the candidate ranking process by building a graph and computing a *random walk* (Spitzer, 1976) over that graph to rank its nodes.

Though these approaches share the same underlying principle, there are differences between them mainly in two aspects: the nature of the nodes to be considered as part of the graph and the nature of the edges. For instance, Gentile et al. (2009) indicate that the nodes represent either concepts (candidates) or features (like words in the title of a certain candidate), and the edges link the candidates with their specific features. Han et al. (2011) define nodes for each anchor to be linked and for its candidates. The edges link each anchor with its candidates but also the candidates among themselves on the basis of their relatedness (see section 3.1.1). In the work of Fernández et al. (2010) the nodes include only the candidates, and the edges are defined on the basis of information about co-occurrence of candidates in Wikipedia articles. A similar approach is used by Ploch et al. (2011), including nodes for the candidates and edges defined on the basis of Wikipedia links.

Note that the PageRank metrics can also be interpreted as a random walk (Page et al., 1999). However, while PageRank, as described in section 3.2.1, operates on the graph of the whole link structure of Wikipedia, the approaches in this section build their own graphs, typically much smaller and tailored to the concrete scenario to be addressed.

The approaches of Gentile et al. (2009) and Han et al. (2011) rely on text-based features to build their graphs: in the work of Gentile et al. (2009) these features are used as nodes in the graph, whereas Han et al. (2011) use a text-based similarity metrics to compute the weights of the edges connecting each anchor with its candidates. Due to this, in the context of this paper we evaluate the approaches of Fernández et al. (2010) and Ploch et al. (2011), which rely only on link information.

As indicated above, Fernández et al. (2010) and Ploch et al. (2011) designed their approaches to link at the same time several anchors in the same context document. Thus, we need to adapt these approaches to the scenario addressed in this paper, where only an anchor a is considered. To do so, each element in $F(d_c)$ is treated as a single-element pseudo-candidate set for an anchor a_i in d_c with $i = 1, \ldots, |F(d_c)|$.

To compute the score for each candidate $c_i \in C(d_c, a)$ according to Ploch et al. (2011) (that we name $RW_P(c_i, d_c))$) we build a graph having as nodes all the candidates and pseudo-candidates (that is, the elements in $C(d_c, a)$ plus the Wikipedia articles linked in $F(d_c)$). An edge between two nodes appears when there is a link in Wikipedia between the articles represented by the nodes. Once the graph is built, the PageRank algorithm is applied to this graph. The score assigned to each node is its PageRank value.

A similar approach is used in the case of Fernández et al. (2010). Again, the nodes include all the candidates and pseudo-candidates, but in this case the edges represent co-occurrences. In particular, there is an edge from node w_i to node w_j when:

1. There is at least a third Wikipedia article w_k that links both to w_i and w_j, that is, $w_i, w_j \in F(w_k)$. These edges are assigned weights according to:

$$weight_C(w_i \to w_j) = \frac{|B(w_i) \cap B(w_j)|}{|B(w_i)|} \tag{28}$$

2. A direct link exists between w_i and w_j, that is, $w_j \in F(w_i)$. These edges are assigned weights as follows:

$$weight_L(w_i \to w_j) = TF_{ij}IDF_j = \frac{n(w_i, w_j)}{\sum_{\forall w_k \in F(w_i)} n(w_i, w_k)} log \frac{|W|}{|B(w_j)|} \tag{29}$$

When two nodes w_i and w_j match both the conditions above, that is, they are directly linked and they co-occur in a third article w_k, a single edge is created that combines the contributions as follows:

$$weight(w_i \to w_j) = \frac{k_C}{k_C + k_L} weight_C(w_i \to w_j) + \frac{k_L}{k_C + k_L} weight_L(w_i \to w_j) \tag{30}$$

where k_L and k_C are configuration parameters. In this case we will use the values $k_L = 0.55$ and $k_C = 0.25$ as suggested by Fernández et al. (2010).

Once the weighted, directed graph is built, the PageRank is computed for this graph. The score for each candidate $c_i \in C(d_c, a)$, named $RW_F(c_i, d_c)$, is the PageRank value of the candidate node in the graph. When the scores of all the candidates are computed, the candidate with highest score is selected as the best one:

$$\arg\max_{c_i}\{RW_P(c_i, d_c)\} \tag{31}$$

$$\arg\max_{c_i}\{RW_F(c_i, d_c)\} \tag{32}$$

Note that, in all the approaches listed in section 3, it might happen that different members of the candidates set obtain the same weight and, thus, there would be a tie in the ranking. We have used the *most frequently linked* (MFL) algorithm to break these potential ties. This algorithm simply assigns a weight to each candidate according to its total number of incoming links, that is:

$$MFL(c_i, d_c) = \sum_{\forall b_j \in B(c_i)} n(b_j, c_i) \tag{33}$$

4. Comparative Evaluation

This section reports the results of the evaluation of the different approaches described in section 3. It is organized as follows: the experimental setup (Wikipedia dataset, corpora, etc.) used for the evaluation is outlined in section 4.1, whereas section 4.2 reports the quantitative results as well as some analysis and interpretation of these results.

4.1 Experimental Setup

In order to evaluate the approaches described in section 3, we need a set of elements: (1) corpora of *queries* to evaluate the approaches; (2) information about the *Wikipedia link structure* that will be used as input by the different approaches; and (3) adequate *metrics* to measure and compare the performance of each approach. The next sections describe these three elements briefly:

4.1.1 Corpora of Queries

In order to evaluate the different approaches, we need corpora containing link-to-Wikipedia queries. According to the definition of the problem (see section 2) each of the queries in these corpora should provide:

- The *anchor* a that is going to be linked.
- The context document d_c in which the anchor a appears. The links in this document provide the context information used by some algorithms.
- A set of *candidates*, $C(d_c, a)$, with Wikipedia articles that are potential targets for the anchor.

- A *golden standard* that indicates the correct answer (candidate in $C(d_c, a)$ to be ranked at the top) for each query. This golden standard is used to compute the performance of the algorithms evaluated.

In the state of the art, we distinguish different approaches regarding the corpora they use for empirical evaluation. A first approach is to build specific corpora. It is followed by early work (Bunescu & Pasca, 2006; Cucerzan, 2007; Mihalcea & Csomai, 2007), as well as by more recent work (Milne & Witten, 2008b; Nguyen & Cao, 2010; Pilz, 2010). A common approach within this first group is to use as corpus a subset of Wikipedia articles and compare the links suggested by automatic algorithms with those provided by Wikipedia editors (see for instance Bunescu & Pasca, 2006; Cucerzan, 2007; Milne & Witten, 2008b; Nguyen & Cao, 2010 and Pilz, 2010). This methodology has also been used in the context of the INEX Link-the-wiki track (Huang, Xu, Trotman, & Geva, 2008). A second alternative is to use already available corpora, like the TAC/KBP corpus (used for instance in Han & Sun, 2011 and Hachey et al., 2011), or the corpora defined by Cucerzan (2007) (used for instance in Gentile et al., 2009 and Ratinov et al., 2011).

In the context of this paper, we adopt both approaches. In particular, we use the following corpora in our comparative evaluation:

- **Cucerzan** In the work of Cucerzan (2007) the authors use two different corpora, one built from Wikipedia articles and the other from manually annotated MSNBC (MSNBC, 2014) news items. We have used these corpora to build our own. In order to do so, we proceeded as follows:

  1. The documents in the Cucerzan corpora contain a set of pairs {*anchor, Wikipedia article*}, each one representing a potential link-to-Wikipedia query. We select randomly 250 pairs from each of the Cucerzan's corpora. These pairs provide us with the anchor a to be linked and the golden standard (correct answer to the query).
  2. A typical approach among the systems in TAC/KBP to generate the candidate set, $C(d_c, a)$, is to rely on information retrieval techniques (Ji et al., 2011). In this paper we adopt this approach. However, as a difference with the TAC/KBP scenario, where the evaluation involves all the stages of entity linking, we center our evaluation only on the candidate ranking stage. Due to this, we are interested in isolating as much as possible this stage from the potential bad performance of a particular candidate search implementation. That is, we are interested in analyzing the performance of different candidate ranking approaches assuming that the candidate search stage is ideal, in the sense that it always returns the correct candidate among the candidate set. Obviously, there does not exist an ideal candidate searcher. Thus, in practice, we rely on a state of the art search engine (Google) and append the correct answer to the candidate set in case it is not found by the search engine. In particular, we query the Google search engine with the text of the anchor and a *site:en.wikipedia.org* restriction, filtering out from the top-10 Google results the Wikipedia pages not included in the *Main* namespace. In case the correct Wikipedia article to be linked is not

included within the Google result set, it is appended at the end, though this only happens in a very limited number of queries: in the Curcerzan Wikipedia corpus the correct candidate was added in 9 out of 250 queries (3.6%), while in the Curcerzan news corpus it was added in 14 out of 250 queries (5.6%).

3. Finally, from the rest of the pairs {*anchor, Wikipedia article*} included in the document where the query has been selected, we obtain the *Wikipedia article* component to be used as context information (links in $F(d_c)$), filtering out the links to articles that are included in the candidate set in order to avoid bias.

- **Ad-hoc corpora** Two ad-hoc corpora have been used in the evaluation. One corpus (**Wikipedia random** corpus) was built by following the methodology suggested by previous works in the state of the art, that is, selecting a set of 500 Wikipedia articles using the *Random article* page (Wikipedia, 2014a).

  The second ad-hoc corpus (**Wikinews** corpus) was built using documents from the English Wikinews site (Wikinews, 2014b). These documents represent news items. They are usually annotated by human editors with Wikipedia links. In this case 500 news items were selected with the *Random article* functionality of Wikinews (Wikinews, 2014a).

  For each document in the total set of 1000 documents in the two ad-hoc corpora, we built a link-to-Wikipedia query by using the following procedure:

  1. A Wikipedia link is randomly selected from the document.
  2. From the selected link we obtain the anchor a and the golden standard (link destination in Wikipedia).
  3. We use the anchor and Google search engine to build the candidate set, as it was indicated in the case of the Cucerzan corpora. Again, we append the correct candidate when it is not included in the Google result set. In particular, the correct candidate was added in 14 out of 500 queries (2.8%) in the case of the Wikinews corpus and in 36 out of 500 queries (7.2%) in the Wikipedia random corpus.
  4. The query context information is obtained from the rest of the links in the document, filtering out those linking to members of the candidate set in order to avoid bias.

- **TAC2010** The TAC 2010 dataset includes a total of 2250 entity linking queries and, for each one, provides the anchor a to be linked, the context document d_c and the golden standard. We have used this dataset as basis to build the last corpus involved in our evaluation. In order to do so, we proceeded as follows:

  1. From the total set of 2250 queries, 1230 have as golden standard the *NIL* answer, that is, there is no Wikipedia article to link in these cases. Thus, no correct candidate instance exists and, due to this, it is difficult to take advantage of these queries to evaluate the candidate ranking process. Taking this into account, we discard these queries and keep the remaining 1020.

2. The documents in the TAC 2010 corpus do not contain links. Thus, we use the following procedure in order to obtain the context links needed by some algorithms:
   - For each query, we analyze its context document d_c by using natural language processing techniques. In particular, we extract named entities (persons, locations and organizations) from text using the Stanford NER tool (Finkel, Grenager, & Manning, 2005).
   - Then, we link the detected entities to Wikipedia using Google. In particular, we query the Google search engine with the text of the named entity and a *site:en.wikipedia.org* restriction, filtering out from the top-10 Google results the Wikipedia pages not included in the *Main* namespace, and assigning as link the top result in the filtered list. We discard the named entities where no Google results are found. The links from named entities to Wikipedia defined with this procedure are used as context information for candidate ranking, filtering out from the context those links to members of the candidate set in order to avoid bias.

   We discard those queries where no context information is available, that is, where the NER tool does not find named entities in d_c, or when they are filtered in the process of linking them to Wikipedia. This results in a total of 1012 valid queries.
3. The candidate set $C(d_c, a)$ for each query is obtained by using the same procedure as in previous corpora: using Google to search the anchor a and appending the correct candidate in case it is not found (which happens in 70 out of 1012 queries (6.9%)).

To summarize, we carry out our evaluation in five different corpora (Cucerzan news, Cucerzan Wikipedia, Wikipedia random, Wikinews and TAC 2010)[1], which add up to a total of 2512 link-to-Wikipedia queries. Boxplot diagrams representing the distributions in each corpus of the number of candidates ($|C(d_c, a)|$) per query, and the number of links ($|F(d_c)|$) per query, are shown in Figures 1 and 2 respectively.

Note that appending the correct candidate to the candidates set was needed in only 143 of the 2513 queries. This indicates that Google performs quite well as a candidate searcher in our case, with a candidate recall near to 95% (considering only the first 10 results). To put this result in context, we can indicate that Hachey et al. (2013) compare several candidate search approaches in the TAC 2009 dataset and report that their candidate recall is below the 75% when limited to a maximum of 10 results. However our results are similar to those by Lehmann et al. (2010), where the authors use Google search combined with a set of additional techniques and report a 97% candidate recall in the TAC 2009 dataset.

4.1.2 Wikipedia Link Structure

All the approaches described in section 3 require information from the Wikipedia link structure to carry out the candidate ranking process. In our case, that information has been

1. These corpora are available to download at: http://www.it.uc3m.es/berto/link-to-wikipedia/survey/

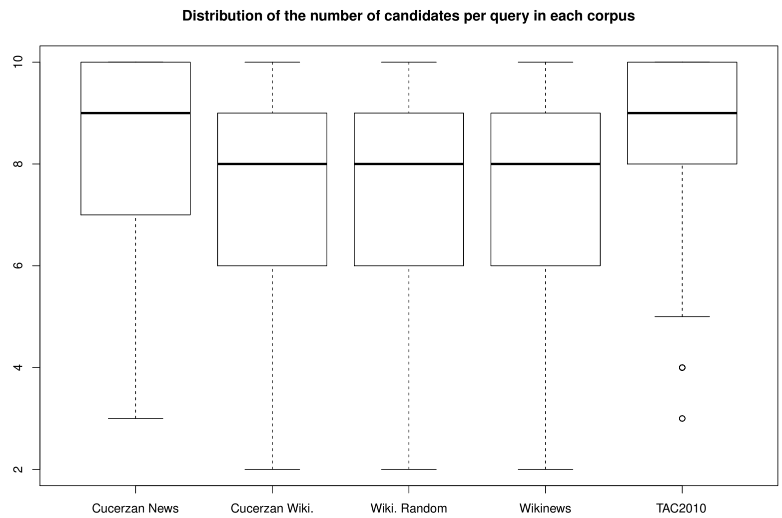

Figure 1: Boxplot diagram of the number of candidates ($|C(d_c, a)|$) per query for each corpus.

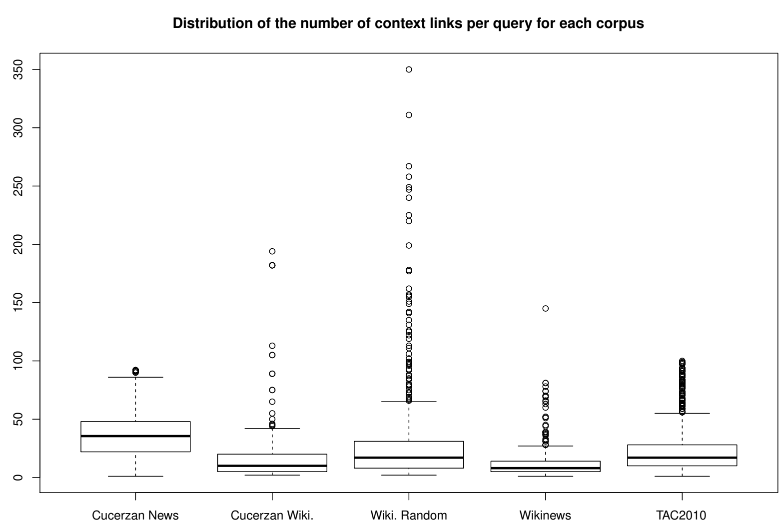

Figure 2: Boxplot diagram of the number of links in the context ($|F(d_c)|$) per query for each corpus.

obtained from a dump of Wikipedia page links provided by DBpedia (Bizer et al., 2009) version 3.8[2], which was generated from a full Wikipedia dump dated in June 2012.

The links dump was preprocessed as follows:

- The redirections were resolved, using the redirections mapping table from DBpedia 3.8[3].
- As indicated in section 2, we consider only the Wikipedia pages that belong to the *Main* namespace. Thus, the links from/to pages in other namespaces (like Talk pages, User pages, etc.) were removed.
- Using the information provided by the disambiguation map from DBpedia 3.8[4], the links from/to disambiguation pages were also removed.
- The inner links (from an article to itself) were also filtered out.
- The evaluation corpora described in section 4.1.1 include in some cases Wikipedia articles. To separate the input data from the evaluation data, all the links with source or destination in one of the Wikipedia articles included in the evaluation corpora were filtered out.

4.1.3 Performance Metrics

To measure and compare the performance of each of the considered approaches, we need adequate metrics. A well-known evaluation metrics for link-to-Wikipedia approaches is the *accuracy*, used for instance by Bunescu and Pasca (2006), Cucerzan (2007), Hachey et al. (2011), Ratinov et al. (2011) and Hachey et al. (2013). The accuracy can be computed as the percentage of queries where the candidate selected by the algorithm is the correct one, or, more formally:

$$Accuracy = \frac{1}{|Q|} \sum_{\forall q \in Q} S(q) \tag{34}$$

Where Q represents a set of evaluation queries, q a particular query in the set, and $S(q)$ a function so that $S(q) = 1$ if the candidate article ranked at the top for query q is the correct answer or $S(q) = 0$ otherwise.

However, as we center the evaluation in the candidate ranking stage of the link-to-Wikipedia task, the accuracy presents a limitation: it does not take into account the actual position of the correct answer within the ranking produced by each algorithm. For example, if one algorithm ranks the correct answer for a query in the 2nd position, whereas another algorithm ranks it in the 8th position, the contribution from this query to the accuracy is zero in both cases, despite the first algorithm having ranked the correct answer higher.

In scenarios where link-to-Wikipedia approaches work under human-supervision (for instance, if these systems are used within the production process of a news agency (Fernández

2. http://downloads.dbpedia.org/3.8/en/page_links_en.nt.bz2 (April, 2014)
3. http://downloads.dbpedia.org/3.8/en/redirects_transitive_en.nt.bz2 (April, 2014)
4. http://downloads.dbpedia.org/3.8/en/disambiguations_en.nt.bz2 (April, 2014)

et al., 2006) to add metadata to news items) the particular order of the candidates is relevant, because in case the top-ranked candidate is not the correct one, the human supervisor can continue reading the ranked list of candidates and select another option. Obviously, the nearer to the top the correct candidate is in the list of suggestions, the better.

Taking this into account, we decided to report performance using not only accuracy, but also two position-based discounting schemes to measure the overall quality of the ranked list of results:

- The *Mean Reciprocal Rank* (MRR) is used for instance in the evaluation of question answering systems (Voorhees, 1999). The MRR of a set of evaluation queries Q can be computed as:

$$MRR(Q) = \frac{1}{|Q|} \sum_{\forall q \in Q} \frac{1}{r(q)} \tag{35}$$

  Where $r(q)$ represents the position of the correct candidate in the rank for a query $q \in Q$.

- As shown in equation 35, the MRR penalizes the differences in position severely. Taking this into account, we also report the results of the *Discounted Cumulative Gain* at a certain level K (DCG@K), which introduces a smoother penalization with position. The DCG@K can be computed as:

$$DCG@K(Q) = \frac{1}{|Q|} \sum_{\forall q \in Q} \sum_{i=1}^{k} \frac{2^{R(q,i)} - 1}{\log_2(1+i)} \tag{36}$$

  Where Q represents a set of evaluation queries, q a particular query in the set, and $R(q,i)$ the relevance score given to the candidate article in position i for query q. We adopt a binary relevance model and, thus, $R(q,i) = 1$ if the candidate in position i is the correct answer and $R(q,i) = 0$ otherwise. Furthermore, we only consider a single candidate as relevant for each query. Taking this into account, the DCG@K is equivalent to its normalized version, the *Normalized Discounted Cumulative Gain* at K (NDCG@K) (Manning et al., 2008), and equation 36 can be simplified into:

$$DCG@K(Q) = \frac{1}{|Q|} \sum_{\forall q \in Q} f(q,k) \text{ with } f(q,k) = \begin{cases} \frac{1}{\log_2(1+r(q))} & \text{if } r(q) <= k \\ 0 & \text{if } r(q) > k \end{cases} \tag{37}$$

  Where $r(q)$ represents the position of the correct candidate in the rank for query q. As it can be seen in equation 37, the bigger the value of $r(q)$ (that is, the farther away the correct candidate from the top of the rank) the lesser the value of the term added to DCG@K. Note also that DCG@1 would be equivalent to the accuracy as defined in equation 34.

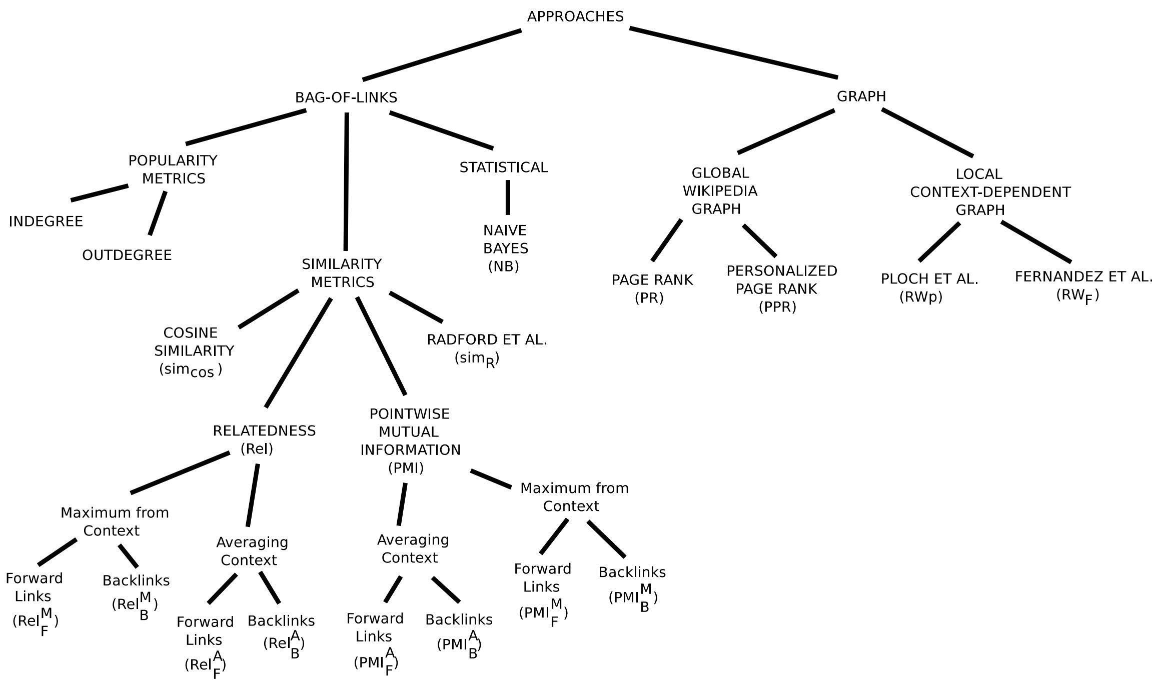

Figure 3: Taxonomy of the different approaches considered for evaluation.

4.2 Evaluation

This section reports the results of the empirical evaluation of the approaches. We have structured the presentation of results in three parts: section 4.2.1 compares the individual algorithms described in section 3, section 4.2.2 analyzes the combination of the different approaches through the use of machine learning techniques and, finally, section 4.2.3 evaluates the impact of changing the search stage on the performance of the algorithms.

4.2.1 Comparison of Individual Approaches

All the approaches described in section 3 (summarized in the taxonomy shown in Figure 3) were evaluated in the corpora described in section 4.1.1. Table 1 reports the accuracy obtained by each approach in the different evaluation corpora. We have highlighted in boldface the best accuracy among the link-based evaluated approaches for each particular corpus.

Table 1 includes a column (*Overall*) that reports the results obtained in the corpus generated by aggregating all the queries. The last column, *Confidence Interval (Overall)*, reports the 95% confidence interval for the accuracy in the *Overall* case, computed using bootstrap methods as suggested by Adibi, Cohen, and Morrison (2004).

As it can be seen in the *Approach* column in Table 1, apart from the approaches considered in section 3, we include the results of two naive algorithms as reference baselines: (1) the *random* algorithm, which simply ranks all the candidates randomly; and, (2) the *most frequently linked* (MFL) algorithm, described in section 3 (see equation (33)). We also report (see row *Google*) the accuracy obtained by using a trivial ranker that simply returns the candidates in the same order as they are defined in the corpus (that is, in the same order as returned by Google, with the correct candidate at the end if it was not found by Google)[5].

Approach	**Cucerzan** News	**Cucerzan** Wiki	**Wiki. Random**	**Wikinews**	**TAC 2010**	**Overall**	**Confidence Interval (Overall)**
Random	0.133	0.149	0.153	0.155	0.118	0.137	(0.124, 0.151)
MFL	0.700	0.630	0.571	0.676	0.668	0.650	(0.631, 0.668)
Google	0.844	0.883	0.795	0.914	0.748	0.813	(0.798, 0.828)
Rel_F^A	0.568	0.458	0.489	0.436	0.495	0.486	(0.466, 0.505)
Rel_F^M	0.416	0.406	0.365	0.338	0.227	0.313	(0.295, 0.331)
Rel_B^A	0.716	0.651	0.717	0.694	0.738	0.715	(0.696, 0.732)
Rel_B^M	0.576	0.542	0.597	0.526	0.418	0.503	(0.483, 0.522)
PMI_F^A	0.204	0.233	0.289	0.206	0.081	0.174	(0.160, 0.189)
PMI_F^M	0.140	0.229	0.252	0.174	0.055	0.144	(0.130, 0.157)
PMI_B^A	0.272	0.321	0.421	0.318	0.237	0.301	(0.283, 0.319)
PMI_B^M	0.192	0.285	0.383	0.274	0.167	0.245	(0.228, 0.262)
sim_{cos}	0.440	0.486	0.483	0.516	0.421	0.461	(0.441, 0.480)
sim_R	0.760	0.687	0.687	0.734	0.725	0.719	(0.701, 0.736)
$indegree$	0.700	0.647	0.581	0.670	0.671	0.653	(0.634, 0.671)
$outdegree$	0.532	0.462	0.327	0.472	0.579	0.491	(0.471, 0.510)
NB	0.772	**0.719**	**0.729**	**0.752**	**0.766**	**0.752**	(0.734, 0.768)
PR	0.684	0.623	0.565	0.668	0.635	0.631	(0.612, 0.650)
PPR	0.692	0.687	0.697	0.702	0.553	0.639	(0.619, 0.657)
RW_P	**0.776**	0.663	0.647	**0.752**	0.732	0.717	(0.698, 0.734)
RW_F	0.760	0.715	0.643	0.744	0.740	0.721	(0.703, 0.738)

Table 1: Accuracy obtained by the different approaches in each of the evaluation corpora.

Figure 4 shows the DCG@K for different values of K in the *Overall* aggregated corpus. MRR values for the different evaluation corpora are reported in Table 2, where, again, we have highlighted in boldface the best MRR among the link-based evaluated approaches for each particular corpus. Furthermore, in order to provide a more detailed idea of the differences among methods, we show in Figure 5 the percentage of queries in which the correct candidate is ranked at position K (with K from 1 to 10) for each algorithm.

We can also provide some empirical results about the run-time of the different algorithms. In particular, the average run-time per query (in seconds) measured on a Linux 2.6.32, Intel Core i7 2.80GHz PC with 16GB RAM was under one second for all the approaches except RW_F and PPR, which run closer to 4 and 571 seconds per query respectively. The relatively large response time of PPR is due to the fact that this algorithm uses context information to personalize the PageRank damping vector. Taking into account that, in general, each query has a different context, this means that we need to run a PageRank

5. Note that in the *Google* case, the queries where the correct candidate is appended to the result set are accounted as errors when computing accuracy.

Approach	Cucerzan		Wiki. Random	Wikinews	TAC 2010	Overall
	News	Wiki				
Random	0.347	0.370	0.374	0.379	0.326	0.353
MFL	0.806	0.763	0.726	0.794	0.791	0.778
Google	0.894	0.922	0.858	0.939	0.828	0.872
Rel_F^A	0.727	0.651	0.667	0.635	0.688	0.673
Rel_F^M	0.607	0.600	0.573	0.548	0.473	0.534
Rel_B^A	0.830	0.787	0.830	0.820	0.847	0.831
Rel_B^M	0.727	0.710	0.746	0.705	0.643	0.691
PMI_F^A	0.428	0.458	0.501	0.435	0.318	0.403
PMI_F^M	0.356	0.440	0.463	0.403	0.271	0.361
PMI_B^A	0.500	0.543	0.622	0.547	0.490	0.534
PMI_B^M	0.415	0.502	0.570	0.503	0.414	0.472
sim_{cos}	0.650	0.671	0.666	0.699	0.620	0.653
sim_R	0.850	0.813	0.805	0.837	0.838	0.830
indegree	0.806	0.773	0.729	0.793	0.794	0.780
outdegree	0.700	0.648	0.550	0.655	0.730	0.668
NB	0.859	**0.834**	**0.832**	**0.848**	**0.861**	**0.850**
PR	0.799	0.76	0.719	0.786	0.774	0.767
PPR	0.812	0.801	0.810	0.810	0.734	0.778
RW_P	**0.864**	0.796	0.778	0.847	0.837	0.826
RW_F	0.854	0.821	0.774	0.837	0.836	0.824

Table 2: MRR obtained by the different approaches in each of the evaluation corpora.

computation on the whole Wikipedia graph for each query in the corpus, a process that is time consuming[6]. Note, however, that we have used a Python implementation which was not optimized and, thus, these results are provided only as a reference.

To contextualize the results reported in Table 1, we can indicate that the TAC 2010 corpus that we are using in our evaluation is practically equivalent (apart from 8 queries removed due to the lack of context information, as indicated in section 4.1.1) to the Non-NIL queries in the TAC 2010 dataset. Due to this, the results reported in the column *TAC 2010* of Table 1 can be roughly compared (less than 1% of error) with the TAC 2010 Non-NIL accuracy reported by some papers in the state of the art. For instance, the best performing approach in TAC 2010 (Lehmann et al., 2010) reported an accuracy on the Non-NIL queries of 80.6%. Note, however, that we have to be cautious with these comparisons, as the results we are reporting would be equivalent to those obtained with an end-to-end system using an ideal candidate search stage (we always append the correct candidate) and without a candidate selection process (we report the results of the candidate ranking stage).

Analyzing the results reported in Table 1, a first conclusion that may be drawn is that the overall accuracy achieved by using the Google ranking is better than that obtained by any of the evaluated approaches. However, if we observe the results obtained for each

6. According to Bianchini, Gori, and Scarselli (2005), this computation depends linearly on the number of edges on the Wikipedia graph.

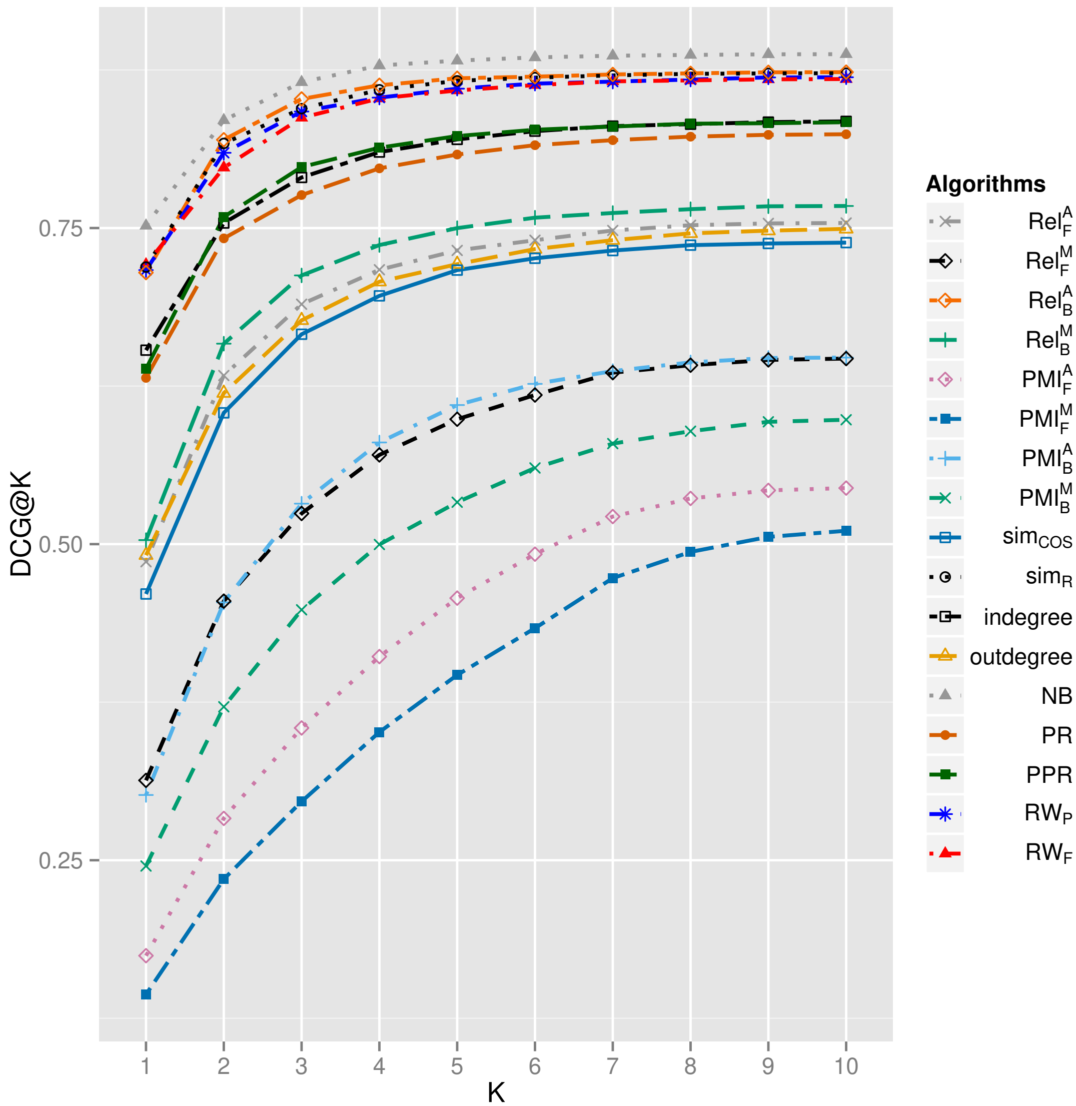

Figure 4: DCG@K values for the different algorithms considered when evaluated on the *Overall* corpus generated by aggregating all the queries.

individual corpus, we can also note that using Google is not always the best approach. In particular, the accuracy of NB in the TAC 2010 corpus is slightly better than that achieved by Google.

To interpret these findings, we have to take into account that previous work in the area (notably that of Chang et al., 2010) had already pointed out that using Google produces relatively good results for the entity linking task (accuracy near 78% in TAC 2009 experimental setup). In that sense, the overall performance obtained by Google is not completely unexpected. The degradation in performance in the TAC 2010 case is partially explained

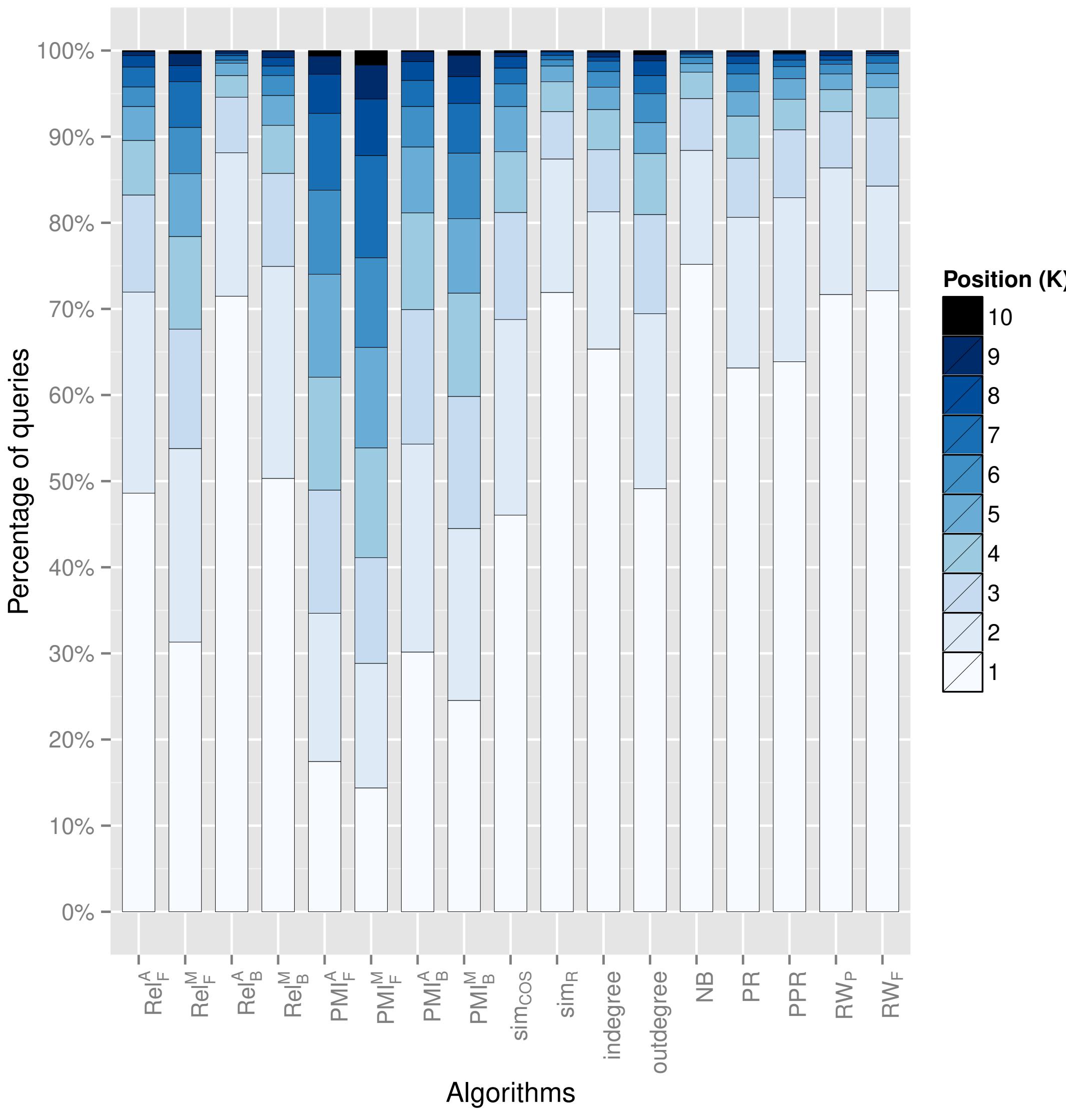

Figure 5: Percentage of queries where the correct candidate is ranked at position K (with K from 1 to 10) for each of the different algorithms compared.

by the fact that this corpus is specifically built for the entity linking task by using a careful targeted process[7]. Due to this, the queries in the TAC 2010 corpus are expected to be challenging. For instance, a common case[8] within this corpus is to have groups of queries sharing the same anchor to be linked, but with different correct answers depending on the particular query. In this case, as the queries share the anchor, they also share the Google ranking and, thus, the top ranked candidate but, as the correct answer changes between queries, using always the Google top ranked candidate as answer introduces some errors. Note also that by using Google we are not taking advantage of the context information, which is expected to be valuable to decide the best link for an anchor, especially when the queries are challenging.

A second conclusion is that naive popularity metrics like the *indegree* (or the MFL, whose performance is very similar to the *indegree*) produce reasonably accurate results. This aspect is consistent with previous work in the state of the art (Ji & Grishman, 2011) where the authors indicate that naive candidate ranking approaches based on web popularity can achieve accuracies around 71% on the TAC 2010 corpus.

Another aspect to be highlighted is that, consistently across evaluation corpora, the *indegree* metrics produces better accuracy than the *outdegree*, which indicates that the number of backlinks offers a better representation of the popularity of a Wikipedia article than the number of its outgoing links. One aspect that may explain, at least partially, this difference in performance is the fact that the number of outgoing links may be high due to several reasons. When a Wikipedia article is long (which indicates that it has received extensive attention by Wikipedia contributors and is, in that sense, popular) we can expect it to have more links than shorter articles. However, there are other cases in which a Wikipedia article can contain many outgoing links. It is the case, for instance, of articles that represent a hub of links, such as *List* articles.

To test the hypothesis that the *outdegree* metrics introduces bias to favor hubs like *List* articles, we compared the number of queries where the candidate that is ranked at the top by *indegree* and *outdegree* is a list (its title starts with *List_*) in the different corpora. These results are shown in Table 3, where x, is the proportion of queries where the candidate ranked at the top is a list, whereas y is the proportion of the queries where the candidate ranked at the top is a list and this is not the correct answer. That is:

$$x = \frac{\text{Queries where the top ranked candidate is a list}}{\text{Total number of queries in the corpus}} \tag{38}$$

$$y = \frac{\text{Queries where the top ranked candidate is a list and is not correct}}{\text{Queries where the top ranked candidate is a list}} \tag{39}$$

As it can be seen in Table 3, the *outdegree* metrics ranks list pages more frequently at the top than the *indegree* metrics. It can also be seen that, in most cases, when the candidate ranked at the top is a list, it is not the correct answer. This particularity explains a significant part of the difference between the overall results of *indegree* and *outdegree*.

7. As indicated in the TAC KBP 2010 task definition document, available at: http://www.it.uc3m.es/berto/link-to-wikipedia/survey/KBP2010_TaskDefinition.pdf (April, 2014)
8. We have identified a total of 144 queries (approximately a 14%) following this pattern.

Approach		Cucerzan		Wiki. Random	Wikinews	TAC 2010	Overall
Name	Param	News	Wiki				
indegree	x	0.004	0.004	0.022	0.006	0.014	0.012
	y	1.0	1.0	0.818	0.667	1.0	0.903
outdegree	x	0.06	0.088	0.138	0.078	0.049	0.078
	y	1.0	1.0	0.956	0.974	1.0	0.979

Table 3: Comparison between *indegree* and *outdegree* regarding the tendency to rank a Wikipedia list at the top.

The difference in performance between using backlinks and forward links can also be noticed in the similarity metrics, where those approaches relying on backlink information (Rel_B^A, Rel_B^M, PMI_B^A, PMI_B^M) produce better results than the corresponding metrics working on forward links (Rel_F^A, Rel_F^M, PMI_F^A, PMI_F^M).

According to the results in Table 1, it can also be pointed out that, among the link-based approaches being evaluated, taking advantage of context information is, in general, beneficial. To support this conclusion we can compare the results of MFL and NB. Note that NB uses the prior $P(c_i)$ (see equations (22) and (23)), which is basically a normalized version of MFL. However, NB combines this prior with the probabilities $P(f_j/c_i)$, which capture context information. As it can be seen in Tables 1 and 2, the result of this combination is that NB produces better results than MFL. Note also that none of the alternatives which use only popularity information are included among the top-5 link-based evaluated approaches with higher accuracy (NB, RW_F, sim_R, RW_P, Rel_B^A). However, using context information is not a sufficient condition to ensure a good performance, as reflected by the results of the PMI variants.

Another conclusion that can be reached is that, in the cases of *relatedness* and *PMI* metrics, averaging the pairwise similarities between the candidate and the articles in $F(d_c)$ (as is done in Rel_B^A, PMI_B^A, Rel_F^A and PMI_F^A) produces, consistently across corpora, better accuracy than relying on the maximum (as is done in Rel_B^M, PMI_B^M, Rel_F^M and PMI_F^M). A possible explanation to this result is that by relying on the maximum similarity we just take into account one of the elements in $F(d_c)$ (the one that maximizes similarity) to represent the semantics of the document d_c, whereas, when averaging, all the elements in $F(d_c)$ contribute to the final similarity value. It is reasonable to think that the set of forward links in d_c provides a more accurate representation of the semantics of the context document than a single link in the document.

With the objective of testing this intuition, we implemented two new variants of Rel_B^M and PMI_B^M, that we name $Rel_B^M(P)$ and $PMI_B^M(P)$. In order to obtain the $Rel_B^M(P)(c_i, d_c)$ scores we compute the $sim_{R_B}(c_i, f_j) \, \forall f_j \in F(d_c)$ as in equation (7). However, instead of just selecting the maximum value, as it is done in equation (7), we select a certain percentage P of the top values and average them. For instance, if $|F(d_c)| = 10$ and $P = 50\%$, we select the top 5 $sim_{R_B}(c_i, f_j)$ values and average them. Note that, with this approach, when $P = 100\%$ the scores would be equivalent to those obtained with Rel_B^A. To obtain the $PMI_B^M(P)(c_i, d_c)$ scores we proceed in a similar way, but using the $P_B(c_i, f_j)$ values (see equation (11)) instead of the $sim_{R_B}(c_i, f_j)$ ones. We evaluated the $Rel_B^M(P)$ and

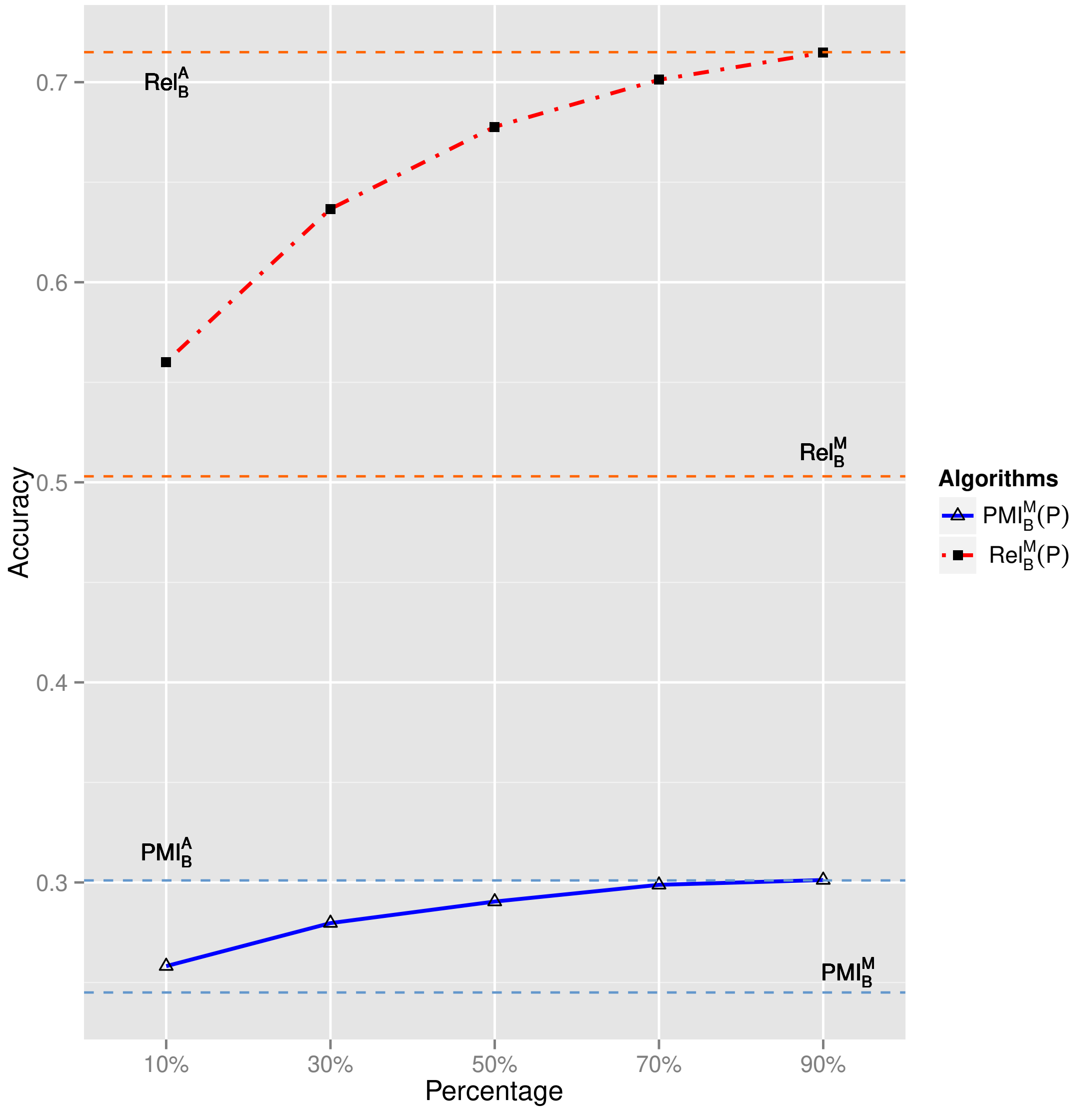

Figure 6: Accuracy values for the $Rel_B^M(P)$ and $PMI_B^M(P)$ approaches for different values of percentage P when evaluated on the *Overall* corpus generated by aggregating all the queries.

$PMI_B^M(P)$ variants on the overall aggregated corpus for different values of percentage P. Figure 6 reports the accuracy obtained by these new variants. We have also included as references horizontal lines representing the overall accuracy of Rel_B^M, PMI_B^M, Rel_B^A and PMI_B^A. As it can be seen, increasing the context information improves results.

Also related with the *relatedness* and the *PMI* metrics is the fact that the results obtained by *PMI* variants are quite poor when compared with the equivalent *relatedness* variants. As an example, see the difference in overall accuracy between Rel_B^A (0.715) and PMI_B^A (0.301).

An inspection of the results of PMI revealed that, because of using absolute values instead of logarithmic values (as in *relatedness*), the *PMI* is more sensitive to outliers. In order to verify and quantify this observation, we decided to compare the results of *PMI* with two other alternatives:

- We implemented a logarithmically smoothed version of the averaging *PMI* variants by adapting equations (8) and (9) as follows:

$$logPMI_F^A(c_i, d_c) = \frac{1}{|F(d_c)|} \sum_{\forall f_j \in F(d_c)} log[P_F(c_i, f_j)] \tag{40}$$

$$logPMI_B^A(c_i, d_c) = \frac{1}{|F(d_c)|} \sum_{\forall f_j \in F(d_c)} log[P_B(c_i, f_j)] \tag{41}$$

- We used the symmetric conditional probability (*SCP*), introduced by da Silva and Lopes (1999). The *SCP* of two Wikipedia documents w_i, w_j can be computed as:

$$S_B(w_i, w_j) = \frac{|B(w_i) \cap B(w_j)|^2}{|B(w_i)||B(w_j)|} \tag{42}$$

  That can also be adapted to use forward links as:

$$S_F(w_i, w_j) = \frac{|F(w_i) \cap F(w_j)|^2}{|F(w_i)||F(w_j)|} \tag{43}$$

  Using equations (42) and (43) the following two metrics were implemented:

$$SCP_F^A(c_i, d_c) = \frac{1}{|F(d_c)|} \sum_{\forall f_j \in F(d_c)} [S_F(c_i, f_j)] \tag{44}$$

$$SCP_B^A(c_i, d_c) = \frac{1}{|F(d_c)|} \sum_{\forall f_j \in F(d_c)} [S_B(c_i, f_j)] \tag{45}$$

We run these approaches on all the corpora, and report the results in Table 4 (accuracies) and Table 5 (MRR). As it can be seen comparing the results in Table 4 with those for PMI_F^A and PMI_B^A in Table 1, a significant increase in performance is achieved by using the logarithmically smoothed version of PMI. It can also be seen that the accuracies reported for SCP_F^A and SCP_B^A are better than those for PMI_F^A and PMI_B^A and more similar to the results of the *relatedness* variants Rel_F^A and Rel_B^A, respectively.

4.2.2 Combining Individual Approaches

We want also to explore the possibility of combining the results of the different link-based approaches to test whether better results can be obtained or not. The approach that we follow to combine the alternatives described in section 3 is based on supervised machine learning techniques. In particular, we use a *learning to rank* (Joachims, 2002) method.

Approach	Cucerzan		Wiki. Random	Wikinews	TAC 2010	Overall	Confidence Interval (Overall)
	News	Wiki					
$logPMI_F^A$	0.392	0.357	0.423	0.370	0.396	0.392	(0.372, 0.411)
$logPMI_B^A$	0.552	0.550	0.669	0.590	0.674	0.632	(0.612, 0.650)
SCP_F^A	0.500	0.454	0.441	0.404	0.286	0.378	(0.359, 0.397)
SCP_B^A	0.728	0.634	0.693	0.658	0.550	0.626	(0.607, 0.645)

Table 4: Accuracy obtained by the logarithmically smoothed PMI variants and the SCP-based metrics in each of the evaluation corpora.

Approach	Cucerzan		Wiki. Random	Wikinews	TAC 2010	Overall
	News	Wiki				
$logPMI_F^A$	0.604	0.574	0.623	0.580	0.607	0.601
$logPMI_B^A$	0.735	0.714	0.800	0.752	0.805	0.778
SCP_F^A	0.681	0.648	0.637	0.610	0.545	0.600
SCP_B^A	0.835	0.788	0.815	0.797	0.741	0.781

Table 5: MRR obtained by the logarithmically smoothed PMI variants and the SCP-based metrics in each of the evaluation corpora.

Though several learning to rank algorithms are available in the state of the art (Liu, 2009), we decided to rely on the *ListNet* method described by Cao et al. (2007). Our decision is backed on the results reported by Chen and Ji (2011), where several alternatives are evaluated and compared in the context of the entity linking problem. In particular, we took advantage of the open source implementation of *ListNet* provided by the University of Massachusetts' RankLib package (Van B. Dang, 2014).

Basically, we use the scores returned by the individual approaches in section 3 as features to be taken into account by the *ListNet* algorithm. The values of the features are normalized in the range $[0, 1]$ to avoid any bias that might favor some of them.

We have tested three different combinations of approaches. The first variant (that we name $ListNet_{All}$) combines all the link-based approaches under evaluation (that is, all the algorithms included in Table 1 except *Google* and the naive references MFL and *Random*). The second variant ($ListNet_{Top}$) combines just the top-5 best performing link-based algorithms under evaluation (according to *Overall* accuracy in Table 1, that is, NB, RW_F, sim_R, RW_P, Rel_B^A). Finally, the third case ($ListNet_{Top+Google}$) combines the top-5 best performing link-based algorithms with the Google baseline. In all the cases, we have used the configuration parameters for *ListNet* that are suggested in the RankLib implementation (1500 epochs and a learning rate of 10^{-5}).

In order to report the accuracy, MRR and DCG@K of the *ListNet* variants, we use the results obtained by averaging 10 repetitions of 10-fold cross validation on the particular corpus being analyzed. Table 6 reports the accuracy in the different corpora for the combinations that have been considered, while Table 7 reports the MRR results for the same

Approach	Cucerzan		Wiki. Random	Wikinews	TAC 2010	Overall
	News	Wiki				
$ListNet_{All}$	0.780	0.716	0.742	0.738	0.678	0.705
$ListNet_{Top}$	0.824	0.786	0.769	0.803	0.793	0.797
$ListNet_{Top+Google}$	0.854	0.864	0.850	0.877	0.816	0.846

Table 6: Accuracy obtained when combining approaches in section 3 with *ListNet* in each of the evaluation corpora.

Approach	Cucerzan		Wiki. Random	Wikinews	TAC 2010	Overall
	News	Wiki				
$ListNet_{All}$	0.867	0.834	0.847	0.848	0.812	0.828
$ListNet_{Top}$	0.888	0.879	0.865	0.885	0.882	0.882
$ListNet_{Top+Google}$	0.916	0.930	0.916	0.929	0.894	0.913

Table 7: MRR obtained when combining approaches in section 3 with *ListNet* in each of the evaluation corpora.

combinations. Figure 7 compares the DCG@K achieved by *ListNet* variants in the *Overall* case with those of the top-5 link-based evaluated approaches.

We can compare the accuracy values reported in Table 6 with those in Table 1. In the overall case, the best result is obtained by $ListNet_{Top+Google}$. The $ListNet_{Top}$ combination shows a lower performance than the Google reference, but outperforms NB (the best of the individual algorithms under comparison). Regarding the $ListNet_{All}$ variant, its accuracy is lower than that obtained by both Google and NB. Similar conclusions can also be reached from Figure 7 for the DCG@K metrics in the overall case. These conclusions suggest that some particular combinations of features can have a positive impact on results.

However, we have to be cautious with these results, because, as indicated by Vanwinckelen (2012), repeated cross validation should not be assumed to provide perfectly precise estimates of a model's predictive accuracy. In fact, Vanwinckelen (2012) does not recommend reporting confidence intervals or making significance claims from repeated cross validation. They report that, though popular among researchers, this practice can contribute to misleading interpretations.

4.2.3 Effect of Changes in the Search Stage

As indicated in section 4.1.1, in order to isolate the results of the candidate ranking algorithms being evaluated from the potential bad performance of a particular candidate search implementation, we would need to rely on an ideal candidate search stage, in the sense that it always returns the correct answer among the candidate set. Obviously, there does not exist an ideal candidate searcher. Thus, in practice, to try to mimic this behavior, we have relied on a state of the art search engine (Google) and we have appended the correct answer to the candidate set in case it is not found by the search engine.

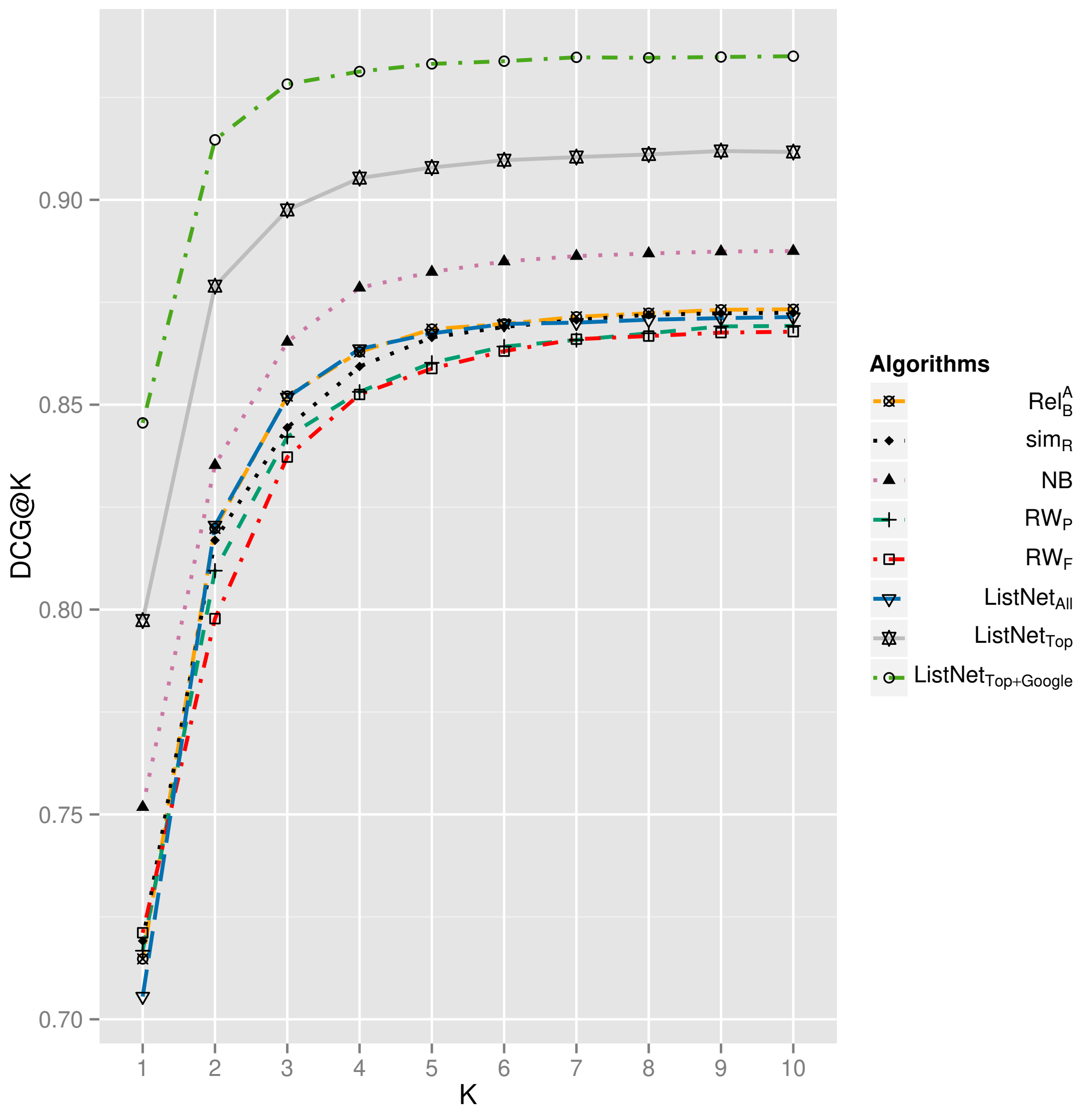

Figure 7: DCG@K values for the top-5 link-based evaluated approaches and *ListNet* variants when evaluated on the *Overall* corpus generated by aggregating all the queries.

However, we are also interested in evaluating the impact on the results achieved by the different algorithms when the aforementioned conditions change into a more restrictive setup. In order to do so, we proceeded as follows:

- We used the information retrieval library Apache Lucene (Apache Software Foundation, 2014) to build an index with the titles of the DBpedia 3.8 pages. Each title was processed by the StandardAnalyzer of Lucene.
- For each of the 1012 queries in the TAC 2010 corpus we carried out the following process:
  - The anchor a of the query is used to search into the Lucene index for the candidates, $C(d_c, a)$. The result set was limited to the top-10 entries, like in previous experiments. However, on the contrary to our previous experiments, when Lucene does not return the correct answer within its result set, we do not append it.
  - As the documents in the TAC 2010 corpus do not contain context links, these are automatically generated using a similar approach to the one described in section 4.1.1: the named entities obtained from the context documents using Stanford NER are resolved into links by querying Lucene with the text of the named entity and assigning as link destination the top result from the search engine (as usual, filtering out the links to articles that are included within the candidate set).

Using the aforementioned procedure we built a new version of the TAC 2010 corpus annotated with Lucene. Thus, we have now two variants of TAC 2010:

- *TAC 2010 Google*, where Google has been used as candidate searcher and the correct candidate is appended to the Google results set in case it is not found. This is the version used in the experiments of previous sections.
- *TAC 2010 Lucene*, which is the version built following the procedure described in this section.

We run all the evaluated approaches, as well as the references *Random* and MFL, in the TAC 2010 Lucene corpus. The accuracies achieved by the different algorithms and their 95% confidence intervals are shown in Figure 8. To ease comparison, we have depicted in the same figure the accuracies for the TAC 2010 Google corpus. We have also included (with the name *Search*) the accuracy achieved when the candidate ranking provided by the search engine (either Google or Lucene) is directly used.

A first aspect to be noted from the results reported in Figure 8 is that, not surprisingly, the accuracies obtained by the different approaches when using Lucene search are, in general, lower. Note that in the Lucene case we are not including the correct candidate in the candidates set. Thus, there are many queries (363 cases, almost a 36% of the total queries) where it is impossible for the candidate rankers to rank the correct candidate at the top.

Another issue to be highlighted is that, if we look at the top-5 best performing link-based approaches in Table 1: NB, RW_F, sim_R, RW_P, Rel_B^A, all of them have greatly reduced their

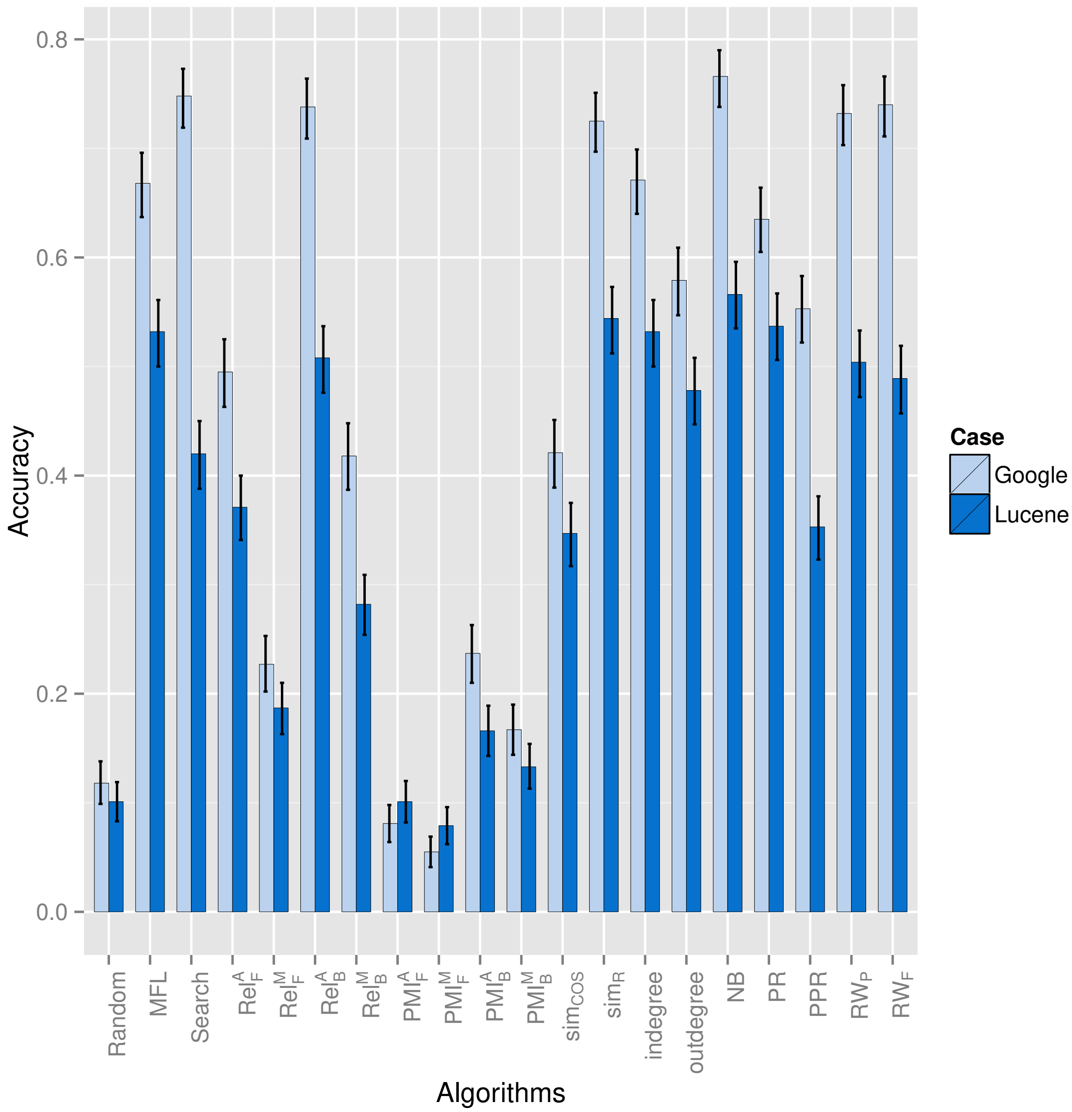

Figure 8: Accuracy obtained by the different approaches on the TAC 2010 Google and TAC 2010 Lucene corpora.

performance in the TAC 2010 Lucene corpus. In fact, though NB is the top performing in that corpus, there is no statistically significant difference with popularity approaches like *indegree* or PageRank (PR). A possible explanation for this result is that in the TAC 2010 corpus the context information is automatically generated from the search engine and not supervised. Thus, we can expect it to be noisy. This noise affects all NB, RW_F, sim_R, RW_P, Rel^A_B, which rely on context information to take their decisions, but does not impact *indegree* and PR, which do not rely on context information. Note that, though the noise in the context information affects both the cases where Google and Lucene are used as search engines, the better performance of Google (note results for *Search*) makes it much worse in the Lucene case.

5. Related Work

The foundations of the link-to-Wikipedia task can be found in two different research communities. First, this area is related with traditional *computational linguistics* tasks like cross-document co-reference resolution (Bagga & Baldwin, 1998), or word sense disambiguation (Navigli, 2009). The main difference with respect to these traditional tasks is that Wikipedia is used as a source of knowledge instead of lexicons such as WordNet (Miller, 1995) typically used in former work (see for instance Li, Szpakowicz, & Matwin, 1995). Second, the link-to-Wikipedia task is also related to the link prediction task in *link mining* (Getoor & Diehl, 2005), though in this case the goal is mainly to decide whether two objects (for instance, two actors in a social network, or an actor and an event) are linked or not, instead of finding the best link destination in Wikipedia for a particular anchor in a text document.

Traditionally, the works of Bunescu and Pasca (2006) and Cucerzan (2007) have been considered as seminal in this area. Since the publication of these papers the problem of linking anchors in a text document to Wikipedia articles has been addressed by several other works, like those referenced in section 3.

Two initiatives are also especially relevant in this sense: the Knowledge Base Population (KBP) track at the Text Analysis Conference (TAC), and Link-the-Wiki track of the Initiative for the Evaluation of XML retrieval (INEX). Both initiatives share the same goal: offer a common environment (corpora, performance metrics, etc) to allow a fair comparative evaluation of different techniques and, thus, foster this area of research. However, as indicated in the introductory section, they approach the link-to-Wikipedia task with slight differences. In the case of KBP, the final aim is to automatically populate a Knowledge Base (KB) built from Wikipedia with information about named entities. Thus, the link-to-Wikipedia variant (*named entity linking*) is focused on these entities and covers the case where no good Wikipedia target exists for the link, as this case indicates the need to add a new entry to the KB. In the case of the Link-the-Wiki INEX track, the focus is set in keeping the links up to date in a rich and dynamic hypermedia document collection (such as a wiki). Therefore, the link-to-Wikipedia variant (*wikification*) covers both common terms and named entities as anchors to be linked, and does not pay special attention to the case where no good Wikipedia target exists, as in this case no link needs to be created.

In both cases, the overview papers published by the organizers of these events (Huang et al., 2008, 2009, 2010; Ji et al., 2010, 2011; Ji & Grishman, 2011) offer a good source of references in this area. However, the comparisons provided by these works refer only to systems taking part in TAC/KBP or INEX, and not to other external work. Furthermore, as indicated in the introductory section, link-to-Wikipedia systems combine, in general, a variety of techniques and features of different types (based on text, on links, etc.) to address the task. Because the results reported in the overviews refer usually to full systems, it is difficult to analyze and compare the performance of the individual techniques that are part of these systems. The goal of our work is doing this analysis and comparison for link-based techniques.

Other surveys related with the task of link-to-Wikipedia and, thus, relevant for the purposes of this paper are that of Navigli and Lapata (2010), Chen and Ji (2011), and Hachey et al. (2013).

Chen and Ji (2011) evaluate several supervised candidate rankers for named entity linking, and compare them with reference unsupervised approaches: a naive algorithm and three different similarity metrics based on textual features. The main goal of this comparison was to assess which machine learning mechanism (maximum entropy, SVM, SVM^{rank} and ListNet) was the top performing. Thus, the results reported by Chen and Ji (2011) and those in our paper are complementary, because, as we indicated in section 4.2.2, the different approaches analyzed here can be used as features in supervised systems. Obviously, this requires to know which supervised techniques work better (Chen & Ji, 2011), but also to know which link-based techniques are better, that is the goal of our paper.

Hachey et al. (2013) re-implement and compare three different named entity linking systems in the state of the art. However, the main goal of their work was different to ours, as the aim of Hachey et al. (2013) was to analyze the impact of the candidate searching and candidate ranking stages in the final performance of the entity linking system.

Navigli and Lapata (2010) compare several metrics based on graph connectivity, including some that we have also considered in our paper, like PageRank and indegree. However, their work has a different scope to ours: it is centered on a different task (word sense disambiguation), and uses a different data source (WordNet).

To the knowledge of the authors, a previous overview and comparison of different link-based approaches for candidate ranking in link-to-Wikipedia systems, as proposed in this paper, is not available at the time of writing.

6. Conclusions and Future Lines

In this paper we have presented an overview of link-based approaches for candidate ranking in link-to-Wikipedia systems. Apart from this overview, a comparative analysis of the different approaches is also carried out. We have structured this analysis into three parts:

- The first part was devoted to compare the performance of the individual approaches according to three metrics (accuracy, DCG@K and MRR) in five different corpora (Cucerzan news, Cucerzan Wikipedia, random Wikipedia articles, random Wikinews articles and TAC 2010). The results in this part of the analysis indicate that, though naive approaches based only on the popularity of the candidates perform reasonably well, taking advantage of the context information is, in general, beneficial in link-based approaches. We have also found that by using information from backlinks we can obtain better results than by using forward links with the same techniques.

- In the second part of the analysis we have combined different approaches by using ListNet. The main conclusion of this part is that, according to the results obtained, combining algorithms can produce positive effects in performance.

- Finally, the third part of the analysis was devoted to evaluate the impact of the candidate search stage in the candidate ranking results, an impact that was found to be very significant.

Regarding potential future lines of development of the work described in this paper, a first aspect to consider is to evaluate the impact of the quality of the context links in the performance of the algorithms. We also want to analyze the effect of ignoring links that might be introducing some noise into the ranking process, like *Lists*. On the opposite case, we are interested in measuring the impact of including links to pages in other namespaces, like *Categories*, that have not been considered in this paper. In this sense, taking *Categories* into account will open the door to the use of semantic relatedness measures based on this information, like those described by Ponzetto and Strube (2007).

According to the results in the paper, using ListNet to combine algorithms can produce positive effects in performance in some cases. However, an exhaustive analysis of different combinations has not been carried out. Thus, another potential line of development could be exploring further combinations of algorithms, either by taking advantage of some proposals of mechanisms for feature selection in learning to rank (Geng, Liu, Qin, & Li, 2007) or empirically.

We have analyzed the different algorithms from the perspective of their performance on the link-to-Wikipedia task. The computational complexity aspects have not been addressed. An exhaustive analysis of the different algorithms along this line is left for future work.

As suggested in section 4.1.3, link-to-Wikipedia systems can be integrated into content production workflows, where they have to interact with human supervisors. Assessing the impact of this human factor on the final performance of the systems can also constitute an area for future research.

Finally, though in this paper we have centered our attention on the candidate ranking stage, link-to-Wikipedia systems usually include other processing stages: identifying the anchors to be linked, searching the candidate links for these anchors, and deciding whether a link is to be suggested or not (detect NIL answers). An end-to-end evaluation including these additional processing stages is also an interesting line to continue the work reported in this paper.

Acknowledgements

In memoriam of Concepción García Alonso (1943-2012) and all the passengers that passed away in the Angrois railway accident (24/Jul/2013).

References

Adibi, J., Cohen, P. R., & Morrison, C. T. (2004). Measuring confidence intervals in link discovery: a bootstrap approach. In *Proceedings of the ACM Special Interest Group on Knowledge Discovery and Data Mining (ACM-SIGKDD-04*.

Apache Software Foundation (2014). Apache Lucene - Welcome to Apache Lucene. Available at: *http://lucene.apache.org/*.

Bagga, A., & Baldwin, B. (1998). Entity-based cross-document coreferencing using the Vector Space Model. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, COLING '98, pp. 79–85, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bianchini, M., Gori, M., & Scarselli, F. (2005). Inside PageRank. *ACM Trans. Internet Technol.*, *5*(1), 92–128.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009). DBpedia - A crystallization point for the Web of Data. *Web Semant.*, *7*(3), 154–165.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Comput. Netw. ISDN Syst.*, *30*(1-7), 107–117.

Budanitsky, A., & Hirst, G. (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Comput. Linguist.*, *32*(1), 13–47.

Bunescu, R. C., & Pasca, M. (2006). Using Encyclopedic Knowledge for Named entity Disambiguation. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics*, EACL.

Cao, Y., Lin, C., & Zheng, G. (2011). MSRA at TAC 2011: Entity Linking. In *Proceedings of the Knowledge Base Population (KBP) track of the 4th Text Analysis Conference (TAC)*. National Institute of Standards and Technololgy (NIST).

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pp. 129–136, New York, NY, USA. ACM.

Chang, A., Spitkovsky, V., Yeh, E., Aguirre, E., & Manning, C. (2010). Stanford-UBC Entity Linking at TAC-KBP. In *Proceedings of the Knowledge Base Population (KBP) track of the 3rd Text Analysis Conference (TAC)*.

Chen, Z., & Ji, H. (2011). Collaborative ranking: a case study on entity linking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pp. 771–781, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The Google Similarity Distance. *IEEE Trans. on Knowl. and Data Eng.*, *19*(3), 370–383.

Cucerzan, S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL 2007*, pp. 708–716.

Cucerzan, S. (2012). The MSR System for Entity Linking at TAC 2012. In *Proceedings of the Knowledge Base Population (KBP) track of the 5th Text Analysis Conference (TAC)*.

da Silva, J. F., & Lopes, G. P. (1999). A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. In *Sixth Meeting of Mathematics of Language*.

Dredze, M., McNamee, P., Rao, D., Gerber, A., & Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pp. 277–285, Stroudsburg, PA, USA. Association for Computational Linguistics.

Erbs, N., Zesch, T., & Gurevych, I. (2011). Link Discovery: A Comprehensive Analysis. In *Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing*, ICSC '11, pp. 83–86, Washington, DC, USA. IEEE Computer Society.

Fader, A., Soderland, S., & Etzioni, O. (2009). Scaling Wikipedia-based Named Entity Disambiguation to Arbitrary Web Text. In *Proceedings of the WikiAI 09 - IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy*, Pasadena, CA, USA.

Fahrni, A., Nastase, V., & Strube, M. (2011). HITS' Cross-lingual Entity Linking System at TAC 2011: One Model for All Languages. In *Proceedings of the Knowledge Base Population (KBP) track of the 4th Text Analysis Conference (TAC)*. National Institute of Standards and Technololgy (NIST).

Fernández, N., Fisteus, J., Sánchez, L., & Martin, E. (2010). WebTLab: A Cooccurence–based Approach to KBP 2010 Entity-Linking Task. In *Proceedings of the Knowledge Base Population (KBP) track of the 3rd Text Analysis Conference (TAC)*. National Institute of Standards and Technololgy (NIST).

Fernández, N., Blázquez, J. M., Fisteus, J. A., Sánchez, L., Sintek, M., Bernardi, A., Fuentes, M., Marrara, A., & Ben-Asher, Z. (2006). NEWS: Bringing Semantic Web Technologies into News Agencies. In *The Semantic Web - ISWC 2006*, Vol. 4273 of *Lecture Notes in Computer Science*, pp. 778–791. Springer Berlin Heidelberg.

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363–370.

Geng, X., Liu, T.-Y., Qin, T., & Li, H. (2007). Feature selection for ranking. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pp. 407–414, New York, NY, USA. ACM.

Gentile, A., Zhang, Z., Xia, L., & Iria, J. (2009). Graph-based Semantic Relatedness for Named Entity Disambiguation. In *Proceeding of the 1st International Conference on Software, Services and Semantic Technologies (S3T)*.

Getoor, L., & Diehl, C. P. (2005). Link mining: a survey. *SIGKDD Explor. Newsl.*, *7*(2), 3–12.

Gracia, J., & Mena, E. (2008). Web-Based Measure of Semantic Relatedness. In *Proceedings of the 9th international conference on Web Information Systems Engineering*, WISE '08, pp. 136–150.

Guo, Y., Tang, G., Che, W., Liu, T., & Li, S. (2011). HIT Approaches to Entity Linking at TAC 2011. In *Proceedings of the Knowledge Base Population (KBP) track of the 4th Text Analysis Conference (TAC)*. National Institute of Standards and Technololgy (NIST).

Hachey, B., Radford, W., & Curran, J. R. (2011). Graph-based named entity linking with Wikipedia. In *Proceedings of the 12th international conference on Web information system engineering*, WISE'11, pp. 213–226, Berlin, Heidelberg. Springer-Verlag.

Hachey, B., Radford, W., Nothman, J., Honnibal, M., & Curran, J. R. (2013). Evaluating Entity Linking with Wikipedia. *Artificial Intelligence*, *194*(0), 130 – 150.

Han, X., & Sun, L. (2011). A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for

Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pp. 945–954, Stroudsburg, PA, USA. Association for Computational Linguistics.

Han, X., Sun, L., & Zhao, J. (2011). Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pp. 765–774, New York, NY, USA. ACM.

Han, X., & Zhao, J. (2009). Named entity disambiguation by leveraging Wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pp. 215–224, New York, NY, USA. ACM.

Haveliwala, T. H. (2003). Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Trans. on Knowl. and Data Eng.*, *15*(4), 784–796.

Huang, D. W., Xu, Y., Trotman, A., & Geva, S. (2008). Overview of INEX 2007 Link the Wiki Track. In Fuhr, N., Kamps, J., Lalmas, M., & Trotman, A. (Eds.), *Focused Access to XML Documents*, pp. 373–387. Springer-Verlag, Berlin, Heidelberg.

Huang, D. W. C., Geva, S., & Trotman, A. (2009). Overview of the INEX 2008 Link the Wiki Track. In Geva, S., Kamps, J., & Trotman, A. (Eds.), *Advances in Focused Retrieval*, Vol. 5631 of *Lecture Notes in Computer Science*, pp. 314–325. Springer Berlin Heidelberg.

Huang, W., Geva, S., & Trotman, A. (2010). Overview of the INEX 2009 Link the Wiki Track. In Geva, S., Kamps, J., & Trotman, A. (Eds.), *Focused Retrieval and Evaluation*, Vol. 6203 of *Lecture Notes in Computer Science*, pp. 312–323. Springer Berlin Heidelberg.

INEX (2014). INEX 2014 main page. Available at: *https://inex.mmci.uni-saarland.de/*.

Ji, H., & Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1148–1158.

Ji, H., Grishman, R., & Dang, H. T. (2011). Overview of the TAC2011 Knowledge Base Population Track. In *Proceedings of the Knowledge Base Population (KBP) track of the 4th Text Analysis Conference (TAC)*.

Ji, H., Grishman, R., Dang, H. T., Griffitt, K., & Ellis, J. (2010). Overview of the TAC2010 Knowledge Base Population Track. In *Proceedings of the Knowledge Base Population (KBP) track of the 3rd Text Analysis Conference (TAC)*.

Jiménez, M., Fernández, N., Fisteus, J., & Sánchez, L. (2013). WikiIdRank++: extensions and improvements of the WikiIdRank system for entity linking. *International Journal on Artificial Intelligence Tools*, *22*(3).

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pp. 133–142, New York, NY, USA. ACM.

Kulkarni, S., Singh, A., Ramakrishnan, G., & Chakrabarti, S. (2009). Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD inter-*

national conference on Knowledge discovery and data mining, KDD '09, pp. 457–466, New York, NY, USA. ACM.

Lehmann, J., Monahan, S., Nezda, L., Jung, A., & Shi, Y. (2010). LCC Approaches to Knowledge Base Population at TAC 2010. In *Proceedings of the Knowledge Base Population (KBP) track of the 3rd Text Analysis Conference (TAC)*.

Li, X., Szpakowicz, S., & Matwin, S. (1995). A WordNet-based algorithm for word sense disambiguation. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, IJCAI'95, pp. 1368–1374, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lin, W.-P., Snover, M., & Ji, H. (2011). Unsupervised language-independent name translation mining from Wikipedia infoboxes. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, EMNLP '11, pp. 43–52, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, T.-Y. (2009). Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.*, *3*(3), 225–331.

Manning, C. D., Raghavan, P., & Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Mihalcea, R., & Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pp. 233–242, New York, NY, USA. ACM.

Miller, G. A. (1995). WordNet: a lexical database for English. *Commun. ACM*, *38*(11), 39–41.

Milne, D., & Witten, I. H. (2008a). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08)*.

Milne, D., & Witten, I. H. (2008b). Learning to link with Wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pp. 509–518, New York, NY, USA. ACM.

MSNBC (2014). MSNBC: news, video and progressive community. Available at: *http://www.msnbc.msn.com*.

National Institute of Standards and Technology (2014a). Text Analysis Conference (TAC). Available at: *http://www.nist.gov/tac/*.

National Institute of Standards and Technology (2014b). Text Analysis Conference (TAC) KBP 2014 Tracks. Available at: *http://www.nist.gov/tac/2014/KBP/*.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Comput. Surv.*, *41*(2), 10:1–10:69.

Navigli, R., & Lapata, M. (2010). An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, *32*(4), 678–692.

Nguyen, H., & Cao, T. (2010). Exploring Wikipedia and Text Features for Named Entity Disambiguation. In Nguyen, N., Le, M., & Swiatek, J. (Eds.), *Intelligent Information and Database Systems*, Vol. 5991 of *Lecture Notes in Computer Science*, pp. 11–20. Springer Berlin / Heidelberg.

Nothman, J., Murphy, T., & Curran, J. R. (2009). Analysing Wikipedia and gold-standard corpora for NER training. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pp. 612–620, Stroudsburg, PA, USA. Association for Computational Linguistics.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. Technical report 1999-66, Stanford InfoLab.

Pilz, A. (2010). Entity Disambiguation using Link based Relations extracted from Wikipedia. In *First Workshop on Automated Knowledge Base Construction (AKBC 2010)*, Grenoble, France.

Ploch, D., Hennig, L., de Luca, E. W., & Albayrak, S. (2011). DAI Approaches to the TAC-KBP 2011 Entity Linking Task. In *Proceedings of the Knowledge Base Population (KBP) track of the 4th Text Analysis Conference (TAC)*. National Institute of Standards and Technololgy (NIST).

Ponzetto, S. P., & Strube, M. (2007). Knowledge derived from Wikipedia for computing semantic relatedness. *J. Artif. Int. Res.*, *30*(1), 181–212.

Radford, W., Hachey, B., Nothma, J., Honnibal, M., & Curran, J. (2010). CMCRC at TAC 2010: Document-level Entity Linking with graph-based re-ranking. In *Proceedings of the 3rd Text Analysis Conference (TAC), National Institute of Standards and Technology, NIST*, Maryland, USA.

Ratinov, L., Roth, D., Downey, D., & Anderson, M. (2011). Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pp. 1375–1384, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sil, A. (2013). Exploring Re-ranking Approaches for Joint Named-entityrecognition and Linking. In *Proceedings of the Sixth Workshop on Ph.D. Students in Information and Knowledge Management*, PIKM '13, pp. 11–18.

Spitzer, F. (1976). *Principles of Random Walk (2nd Edition)*. Springer.

Van B. Dang (2014). RankLib (software package). Available at: *http://people.cs.umass.edu/˜vdang/ranklib.html*.

Vanwinckelen, Gitte; Blockeel, H. (2012). On estimating model accuracy with repeated cross-validation. In *Proceedings of the 21st Belgian-Dutch Conference on Machine Learning*, pp. 39–44.

Voorhees, E. (1999). TREC-8 Question Answering Track Report. In *Proceedings of the 8th Text Retrieval Conference*, pp. 77–82.

Wikinews (2014a). Wikinews Random Page Generator. Available at: *http://en.wikinews.org/wiki/Special:Random*.

Wikinews (2014b). Wikinews, the free news source. Available at: *http://en.wikinews.org/*.

Wikipedia (2014a). Wikipedia Random Page Generator. Available at: *http://en.wikipedia.org/wiki/Special:Random*.

Wikipedia (2014b). Wikipedia:Namespace - Wikipedia, the free encyclopedia. Available at: *http://en.wikipedia.org/wiki/Wikipedia:Namespace*.

Yeh, E., Ramage, D., Manning, C. D., Aguirre, E., & Soroa, A. (2009). WikiWalk: random walks on Wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-4, pp. 41–49, Stroudsburg, PA, USA. Association for Computational Linguistics.