BEL: Bagging for Entity Linking

Zhe Zuo, Gjergji Kasneci, Toni Gruetze, Felix Naumann
Hasso Plattner Institute
Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam, Germany
{firstname.lastname}@hpi.uni-potsdam.de

Abstract

With recent advances in the areas of knowledge engineering and information extraction, the task of linking textual mentions of named entities to corresponding ones in a knowledge base has received much attention. The rich, structured information in state-of-the-art knowledge bases can be leveraged to facilitate this task. Although recent approaches achieve satisfactory accuracy results, they typically suffer from at least one of the following issues: (1) the linking quality is highly sensitive to the amount of textual information; typically, long textual fragments are needed to capture the context of a mention, (2) the disambiguation uncertainty is not explicitly addressed and often only implicitly represented by the ranking of entities to which a mention could be linked, (3) complex, joint reasoning negatively affects the efficiency.

We propose an entity linking technique that addresses the above issues by (1) operating on a textual range of relevant terms, (2) aggregating decisions from an ensemble of simple classifiers, each of which operates on a randomly sampled subset from the above range, (3) following local reasoning by exploiting previous decisions whenever possible. In extensive experiments on hand-labeled and benchmark datasets, our approach outperformed state-of-the-art entity linking techniques, both in terms of quality and efficiency.

1 Introduction

Named-entity linking (NEL) is the task of establishing a mapping from textual mentions of named entities to canonical representations of those entities in a knowledge base. Often, textual mentions are ambiguous; that is, a mention could refer to multiple named entities, but only one of them is correct in the given textual context. Resolving these ambiguities is often referred to as *named entity disambiguation* (NED), which is a highly challenging aspect of an NEL process. More specifically, a robust NEL algorithm has to robustly resolve ambiguities and thus build on robust NED methods. The NED problem, however, is often ill-posed, as only the right context and background knowledge can help disambiguate entities. In many cases, the contextual information is implicit in nature and may be latently spread across various passages or documents, and background knowledge may not be sufficient, which makes the disambiguation task challenging even for human readers. As an example, consider the sentence: "*London spent $80,000 ($2,040,000 in current value) to build a 15,000-square-foot stone mansion ('Wolf House') on the property.*" A human reader knows that in general money is spent by people, but sometimes also city councils can spend money, and hence, in the above sentence "London" may refer to a person or to the city of London. However, when considering the contextual information, especially the key phrase "Wolf House", and the fact that this was the name of the mansion of the writer Jack London, the disambiguation of "London" becomes obvious.

The NED problem is abundant, and the above subtleties place it right at the heart of many artificial intelligence applications, such as semantic search, machine translation, business intelligence, topic detection, text summarization, machine vision, and many more. In the context of information systems, the problem has been addressed in many different flavors and settings, e.g., in the structured setting of

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2075–2086, Dublin, Ireland, August 23-29 2014.

record-linkage and *duplicate detection*, where the goal is find database records that refer to the same named entity (Bhattacharya and Getoor, 2007; Naumann and Herschel, 2010), in the semi-structured setting of cleaning XML data (Weis and Naumann, 2005) or annotating Web tables (Limaye et al., 2010), in the context of enriching Wikipedia information boxes (Wu and Weld, 2008), for the alignment of knowledge bases (Aumueller et al., 2005; Lacoste-Julien et al., 2013), and most prominently, in the setting of Natural Language Processing (Bagga and Baldwin, 1998; Mann and Yarowsky, 2003; Fleischman and Hovy, 2004; Bunescu and Pasca, 2006; Cucerzan, 2007), which is also the setting of this work.

In the latter setting, the proliferation of clean knowledge bases with rich semantic relations between Web entities, e.g., DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), or YAGO (Suchanek et al., 2007), has given rise to novel, reliable NEL techniques (see Section 2) that exploit the semantic relatedness between entities for the linking process (Shen et al., 2012; Hoffart et al., 2011b; Hoffart et al., 2012).

Our disambiguation model builds on a majority-voting strategy that employs a bagging of multiple ranking classifiers, thus the name BEL: Bagging for Entity Linking. Each ranking classifier operates on a randomly sampled subset of terms surrounding the mention in focus. These terms are sampled from a so-called textual *range of relevant terms*, i.e., terms that are most promising for determining the context of the mention. Finally, based on the sampled terms, each ranking classifier proposes a ranked list of candidate entities and the mention is linked to the entity that is proposed as top-ranked candidate by the majority of the classifiers.

In summary, the main contributions of this work are:

1. A novel ensemble-based disambiguation approach that exploits the terms that surround a textual mention to best capture its context; a parsimonious linking model that combines the above method with a prior probability (similar to the one presented in (Fader et al., 2009), (Hoffart et al., 2011b), or (Lin et al., 2012)) of a candidate named entity being referred to by a given mention yields a highly efficient linking process.

2. An analysis of the disambiguation impact of the components used in BEL on the final linking decision.

3. A detailed quality and efficiency comparison with the state-of-the-art methods of Cucerzan (2007), Hoffart et al. (2011b), and Hoffart et al. (2012) on multiple real-world and synthetic datasets; apart from being more efficient, BEL also achieves a linking quality that is comparable to or even better than that of the above methods.

The remainder of the paper is organized as follows: The next section discusses related work. Section 3 is devoted to our NEL approach. The experimental evaluation is presented in Section 4, before we conclude in Section 5.

2 Background and Related Work

There is a vast array of literature on the topic of resolving ambiguous mentions of named entities. We focus on relevant disambiguation strategies for the NEL problem and leave aside natural language processing (NLP) techniques on named entity recognition and part-of-speech tagging; although, admittedly, for the recognition of entity mentions, such techniques are indispensable. In this work, we assume such NLP techniques are given and use the Stanford NER Tagger (Finkel et al., 2005) to reliably recognize textual mentions of named entities. Another field that we bypass is that of *record linkage* or *duplicate detection*, where the focus is on comparing sets of database records and identifying mappings between records referring to the same entity. Obviously, record linkage methods operate on structured data, such as database entries with a predefined set of attributes (commonly with a known value range), which is different from our NLP setting.

In traditional methods, each mention and each named entity is represented by a vector of terms occurring in its textual context. Vector-based similarity measures are applied to capture the affinity between a mention and a named entity. The feature values can go beyond simple unigram terms and consist of compound terms, such as bigrams, key phrases, encyclopedic facts, or categorical descriptions. For example, Pedersen et al. (2005) employed salient bigrams to represent the context of a mention; Mann and Yarowsky (2003) included biographic facts into the vector representation of a named entity, whereas Cucerzan (2007) extended the term-based feature set of a Wikipedia entity by information from other articles linking to it, but instead of using the whole article text, only some key phrases and immediate Wikipedia categories were included. Bunescu and Pasca (2006), after deriving an entity dictionary from Wikipedia, for a given mention, rank entities by a kernel-based similarity between the textual context of the mention and the Wikipedia text and categories of the candidate entity. The mention is linked to the most similar entity.

The disambiguation problem has also been formulated as a probabilistic reasoning problem. For example, Fleischman and Hovy (2004) trained a maximum entropy model to infer the probability that two mentions represent the same entity and used a modified agglomerative clustering algorithm to cluster mentions using the probabilistic similarity measure. Similarly, Sil et al. (2012) used a log-linear model to represent the probability of a named entity being referred to by a mention. For both above methods, the selection of features and efficient strategies for learning their weights are crucial, as ideally all feature weights should be learned in a joint fashion, which can be computationally expensive and is often impeded by the "curse of dimensionality".

Note that many of the above techniques model the implicit relatedness between terms (and term compounds), where the general idea is that two terms are related if many Web pages contain both of them. Measures building on this idea were refined and extended in (Milne and Witten, 2008) and (Huang et al., 2012), especially for the relatedness between Wikipedia articles. Such implicit relatedness can lead to a large candidate space; to effectively prune this space, entity prominence priors have been integrated in various recent disambiguation models, e.g., (Fader et al., 2009; Hoffart et al., 2011b; Lin et al., 2012).

Other techniques model explicit, relationship-based similarities between entities; for example, Du et al. (2013) employed similarity measures that captured the average pair-wise proximity between candidate entities in the knowledge graph, as well as their average pair-wise conceptual similarity by means of the lowest-common-ancestor classes. (Hoffart et al., 2011b; Hoffart et al., 2012) exploited the hypernymy- and key-phrase-based relatedness, between the k candidate entities in the knowledge base, to jointly link k mentions occurring in the same paragraph. A prior probability of a candidate entity being referred to by a mention was combined with the above relatedness measures in an objective maximization function. The intuition behind the hypernymy-based relatedness was that in order for k mentions (that occur in the same textual context) to be linked correctly to $l \leq k$ named entities in the knowledge base, the l entities should jointly exhibit a high "semantic" relatedness, which in (Hoffart et al., 2011b) is referred to as *coherence*. Despite this principled modeling of the NEL problem in (Hoffart et al., 2011b; Hoffart et al., 2012) and the impressive quality results reported in those works, efficiency seems to be the main bottleneck of such collective inference models. We argue that a Web-scale NEL process should avoid complex reasoning strategies wherever possible. Concerns along these lines have been also expressed in (Lin et al., 2012), where the authors highlight the need for the application of NEL techniques at Web scale.

The approach presented in this paper, BEL, avoids complex, coherence-based joint reasoning. It also avoids the processing of long textual passages, where multiple mentions have to occur. Instead, we show that a careful light-weight, independent reasoning on the linking of mentions can lead to a linking quality that is comparable to and sometimes even better than the one achieved by the above methods.

3 The BEL Algorithm

In this work, the focus is not on the recognition of named entity mentions in a text but rather on their disambiguation once the mentions are known. Throughout this work we assume that a reliable named entity recognition tool is available. BEL relies on the Stanford NER Tagger (Finkel et al., 2005) to recognize textual mentions of named entities. Once the mentions have been recognized, BEL retrieves promising candidate entities from the knowledge base and employs a careful, majority-voting algorithm to take the best possible linking decision based on the textual context of the mentions. The method is described in the following subsections.

3.1 High-Level Overview of the BEL Algorithm

Algorithm 1 gives a high-level overview of the BEL approach. The only assumption we make is that the textual corpus from which the knowledge base has been derived is freely available. For example, the textual corpus of knowledge bases such as YAGO or DBpedia is Wikipedia, which is an open source of information about the entities in the two knowledge bases.

Exemplarily, in Algorithm 1, we use the YAGO knowledge base to highlight the main idea of the algorithm. YAGO is a clean knowledge base with structured information about a large proportion of the entities contained in Wikipedia, thus being a popular representative of many state-of-the-art knowledge bases derived from Wikipedia.

Once the set of mentions has been derived from a given document (line 1), for each mention, a list of promising candidates is derived from Wikipedia. The candidates are ranked by a so-called “prominence” score, representing the probability of a Wikipedia article (i.e., the entity represented by the article) being referred to by the mention (lines 2, 3). In case the list of candidates is empty, the corresponding mention is linked to a designated entity, E_{NULL}, meaning that the mention cannot refer to a YAGO entity (lines 4, 5). The same holds for the case that the top-ranked candidate occurs in Wikipedia but not in YAGO (lines 7 - 9). Otherwise, the joint majority decision of multiple bagged ranking classifiers is computed (lines 11 - 13). Only if there is a majority consensus about a candidate (i.e., the candidate is ranked as top candidate by the majority of the classifiers), the mention is linked to that candidate; otherwise, the mention is linked to E_{NULL} (lines 14 - 18).

Algorithm 1 Bagging for Entity Linking Algorithm

```
Input: document file $\mathbf{D} = (t_1, t_2, \ldots)$, HashMap $\mathbf{V}$ that maps the ID of a ranking classifier to the top-ranked candidate entity by that classifier.
Output: linkage between mentions $\mathbf{M} = \{m_1, m_2, \ldots\}$ in $\mathbf{D}$ and corresponding entities $\mathbf{E} = \{e_1, e_2, \ldots\}$ in YAGO.
1:  $\mathbf{M}$ := recognizeMentions($\mathbf{D}$)
2:  for each mention $m_i \in \mathbf{M}$ do
3:      $\mathbf{L}_{m_i}$ := getTopKCandidates($k$,$m_i$) /*according to the “prominence” score $S_{PR}(e, m_i)$*/
4:      if $\mathbf{L}_{m_i}$ is empty then
5:          link $m_i$ to $E_{NULL}$ /*i.e.,mention cannot be linked*/
6:      else
7:          $e' := arg\,max_{e \in L_{m_i}} S_{PR}(e, m_i)$
8:          if $e'$ is not in YAGO then
9:              link $m_i$ to non-YAGO entity $E_{NULL}$
10:         else
11:             for each ranking classifier $S_n$ do
12:                 $\mathbf{V}$.put($n, arg\,max_e(SimScore(e, S_n, m))$)
13:             end for
14:             if an $e^*$ occurs more than $\frac{|\mathbf{V}|}{2}$ in $\mathbf{V}.values()$ then
15:                 link $m_i$ to $e^*$
16:             else
17:                 link $m_i$ to non-YAGO entity $E_{NULL}$
18:             end if
19:         end if
20:     end if
21: end for
```

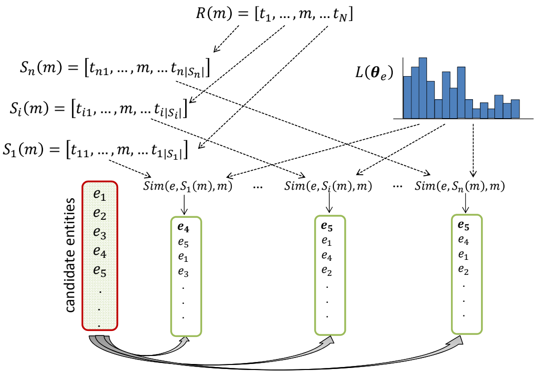

$R(m) = [t_1, \ldots, m, \ldots t_N]$
$S_n(m) = [t_{n1}, \ldots, m, \ldots t_{n|S_n|}]$
$S_i(m) = [t_{i1}, \ldots, m, \ldots t_{i|S_i|}]$
$S_1(m) = [t_{11}, \ldots, m, \ldots t_{1|S_1|}]$
$L(\boldsymbol{\theta}_e)$
$Sim(e, S_1(m), m)$ … $Sim(e, S_i(m), m)$ … $Sim(e, S_n(m), m)$
candidate entities
e_1 e_2 e_3 e_4 e_5
e_4 e_5 e_1 e_3
e_5 e_1 e_4 e_2
e_5 e_4 e_1 e_2

Figure 1: Strategy for generating ranking classifiers, each of which operates on a randomly sampled subset $S_i(m)$ from a set $R(m)$ of relevant terms surrounding m and assigns contextual similarity scores to the candidate entities based on that subset.

In addition, for efficiency reasons, BEL exploits previous disambiguation decisions whenever possible. If a mention occurs multiple times in a document and was already reliably linked b times to the same named entity in YAGO, the previous linking decisions (for that mention) are reused without rerunning the disambiguation process (i.e., in the experimental evaluation of BEL, the default value of b is 2). This heuristics may lead to incorrect linkings, but in empirical evaluations on real-world datasets the algorithm has shown a robust quality behavior, while being highly efficient (see Section 4).

The runtime of the algorithm is dominated by the computation of the contextual similarity scores of each ranking classifier. More specifically, since the parameters needed by each classifier are precomputed, each classifier needs only $O(N \log N)$ steps to propose a context-based ranking of the N candidates derived by the “prominence” score. Since the classifiers operate independently from each other, the algorithm allows parallel computation of the contextual similarity scores. However, in this work we have implemented a sequential version, which for K different ranking classifiers has a complexity of $O(KN \log N)$.

3.2 Bagging of Ranking Classifiers for Majority Voting

The main idea behind BEL is to leverage different contextual representations of a mention. Each such representation is used by one ranking classifier to rank the candidate entities by their similarity to that representation. As shown in lines 14 and 15 of Algorithm 1, the mention is linked to the candidate that is ranked as top entity by the majority of the classifiers. This idea gives rise to several questions: (1) How to derive the different contextual representations of a mention? (2) How to compute the similarity between a contextual representation and a candidate entity? (3) How to prune the candidate space in such a way that only the most promising entities are considered by each of the ranking classifiers?

Obviously, the latter question is focused on efficiency; to address it, we exploit a precomputed index, constructed by exploiting the intra-Wikipedia links and the Wikipedia Redirect Pages. For every mention m, the index contains entities that might refer to it along with a probabilistic "prominence" score $P(e|m)$ by which a candidate entity e might refer to the mention. To compute this "prominence" score, we collect all terms that occur in a Wikipedia article or redirect-page and are hyperlinked to another Wikipedia article. From this collection we derive relative frequencies by which a Wikipedia entity (i.e., specific article) was hyperlinked from a given term. For example, Jordan is hyperlinked to the article about the country of Jordan 60% of the time, 20% of the time it is hyperlinked to the basketball player Michael Jordan, etc. These relative frequencies are estimations of the probability of an entity given a mention $P(e|m)$. Empirically (see also Figure 2a in the evaluation section) we have found out that when ranked by this score, the top-40 candidate entities already yield a overwhelming coverage rate of ≈92.4% for the correct entity. For top-100 this coverage rate increases only marginally (by only ≈0.5%). Thus considering only the top-40 entities in the candidate lists (which in general might contain hundreds or even thousands of entities), is not only an efficient but also an effective pruning strategy.

The second question involves the semantics of the similarity score. Suppose that $S_i(m)$ stands for the i'th contextual representation of the mention m. Our model is probabilistic in nature and holistic in the sense that the above "prominence" score $P(e|m)$ just falls off the model by following principled mathematical derivations. We start by reasoning about the probability of a candidate entity e given the mention m and its context $S_i(m)$:

$$\begin{aligned} P(e|S_i(m), m) &= \frac{P(S_i(m)|m, e)P(m|e)P(e)}{P(S_i(m)|m)P(m)} && (1) \\ &= \frac{P(e|m)P(S_i(m)|m, e)}{P(S_i(m)|m)} && (2) \\ &\propto P(e|m)P(S_i(m)|m, e) && (3) \\ &\propto logP(e|m) + logP(S_i(m)|m, e) && (4) \end{aligned}$$

The last two steps in the above derivation mean that ranking the candidate entities by the similarity score $P(e|m)P(S_i(m)|m, e)$ or by $logP(e|m) + logP(S_i(m)|m, e)$ yields the same ranking as $P(e|S_i(m), m)$. Note that in general $S_i(m)$ depends on the entity e and not on the mention m. Hence, we can estimate $P(S_i(m)|m, e)$ as $P(S_i(m)|e)$. So the final similarity score is given by:

$$Sim(e, S_i(m), m) := logP(e|m) + logP(S_i(m)|e) \quad (5)$$

We estimate $P(S_i(m)|e)$ as the probability of $S_i(m)$ being generated by a language model (Zhai and Lafferty, 2004) on the terms describing e in Wikipedia. Those terms are collected from the Wikipedia article of e after removing stop words. Such a language model is described by means of frequency parameters $\boldsymbol{\theta}_e$. We construct it by indexing the terms and their frequencies in the corresponding Wikipedia articles. Figure 1 depicts the general idea behind our approach. For different contextual representations $S_1(m), ..., S_n(m)$ of a mention m, the ranking classifier responsible for $S_i(m)$ computes $Sim(e, S_i(m), m)$ for each candidate entity e and ranks the candidates by this score. Finally m is linked to the candidate that is ranked as the top entity by the majority of the ranking classifiers. This majority voting strategy reduces the uncertainty of the linking process and leads to higher precision than a single ranking classifier, while still maintaining a high recall.

The final question concerns the computation of the contextual representations $S_i(m)$ of a mention m. We derive such representations by randomly sampling terms that occur in the local vicinity of a mention in the text. More specifically, to generate a contextual representation $S_i(m)$ from a range of N relevant terms around a mention m, we uniformly sample N times with replacement. We run the same procedure for all n representations. This sampling technique is known as bootstrapping (Breiman, 1996) and has been shown to have several advantages over other sampling procedures, such as increasing the contextual diversity and mitigating strong dependencies between features. Indeed, in the experiments, the bagging of the ranking classifiers lead to a significant improvement of $\approx$2.5% in terms of precision compared to the simple case where no bagging is used (see Section 4).

3.3 Recognizing Non-YAGO Entities

For an improved accuracy of the linking process, it is also crucial to reliably recognize true negatives, i.e., mentions that refer to entities that are not present in the underlying knowledge base. In case of the YAGO knowledge base, we first check whether the most prominent Wikipedia entity for a given mention is presented in YAGO; if this is not the case, the mention is classified as a non-YAGO entity. Note that many entities from Wikipedia are not present in YAGO, either due to recently added articles, or to articles that represent concepts[1]. Furthermore, a flexible threshold is used to recognize a non-YAGO entity. It is calculated as the maximum similarity score among the Wikpedia entities in the candidate list that are not present in the knowledge base, or as a default "prominence" score, when there is no such entity. If none of the candidates has a higher score than the threshold, the corresponding classifier proposes E_{NULL} as the best candidate. Also, in the simple case that the retrieved list of candidates is empty, the mention is classified as a non-YAGO entity. Although, these strategies are relatively straight-forward, they lead to a notable improvement in the recognition of true negatives. Further investigation of more elaborate strategies for the reliable detection of true negatives is part of our future work agenda.

3.4 Efficiency Aspects

For a better overview of the key efficiency aspects that are leveraged by BEL, we give here a succinct summary:

- Early pruning of the candidate space while maintaining a high coverage of promising candidates
- Local and independent reasoning strategy based on sliding windows and bootstrapping aggregation for the disambiguation process
- Highly efficient, in-memory processing of randomly sampled subsets
- Previous disambiguation decisions are exploited whenever possible; e.g., for people, locations, or company names that reoccur in a similar form in a document, the disambiguation process is run only once.

As it will be shown in the next section, the above considerations lead to a highly efficient linking process that often outperforms the evaluated state-of-the-art techniques, both in terms of quality and efficiency.

4 Experimental Evaluation

4.1 Datasets

Three datasets were used to evaluate the BEL approach. As a knowledge base for evaluation, we used YAGO2 (Hoffart et al., 2011a).

[1] In YAGO, the concepts have been derived from WordNet.

Table 1: Datasets overall information

	CoNLL-YAGO	CUCERZAN	KORE
articles	76	336	50
mentions (total)	1431	5343	148
mentions (non-YAGO)	279	936	7
word count (avg.)	173	384	12

CoNLL-YAGO: This dataset contains 76 randomly picked Reuters news articles of CoNLL 2003 data (Tjong Kim Sang and De Meulder, 2003). We have manually labeled the mentions, which are recognized by the Stanford NER Tagger (Finkel et al., 2005), to the corresponding entities in YAGO2.

CUCERZAN: This dataset consists of 350 Wikipedia articles that were randomly selected by S. Cucerzan to evaluate his approach (Cucerzan, 2007). The annotated entities in this corpus are named entities derived from the hyperlinks of mentions in these 350 Wikipedia articles. Since some of the articles are not available anymore in the Wikipedia archive, we have recovered 336 out of the 350 articles of the original corpus.

KORE: This small dataset was produced in the realm of AIDA (Hoffart et al., 2012). It is a synthetic corpus consisting of 50 very short articles, where each article contains one or more hand-crafted sentences about different ambiguous mentions of named entities. This dataset is quite difficult, as the named entities in this corpus are ambiguous with sparse context.

4.2 Evaluated Approaches

We compared BEL to three other prominent approaches (Hoffart et al., 2011b; Hoffart et al., 2012; Cucerzan, 2007), which, as reported in the corresponding papers, outperform many state-of-the-art algorithms in terms of disambiguation and linking quality. Experience-wise, we can confirm that the very recent AIDA approaches (Hoffart et al., 2011b; Hoffart et al., 2012) have indeed raised the bar for many entity linking methods. In our experiments, these algorithms showed a highly reliable behavior, even with respect to difficult disambiguation tasks.

The AIDA approach comes in different versions: In its original version (Hoffart et al., 2011b), it exploits a graph-based connectivity between candidate entities of multiple mentions (i.e., graph coherence, e.g., derived from the *type, subclassOf* edges of the knowledge graph or from the incoming links in Wikipedia articles) to determine the most promising linking of the mentions. We refer to this version of AIDA as AIDA-GRAPH. In another version that has been optimized for datasets such as KORE (Hoffart et al., 2012), AIDA's coherence model has been extended to recognize key-phrases for named entities, which are then used to determine a similarity score based on key-phrase overlap between candidate entities. We refer to this version as AIDA-KORE.

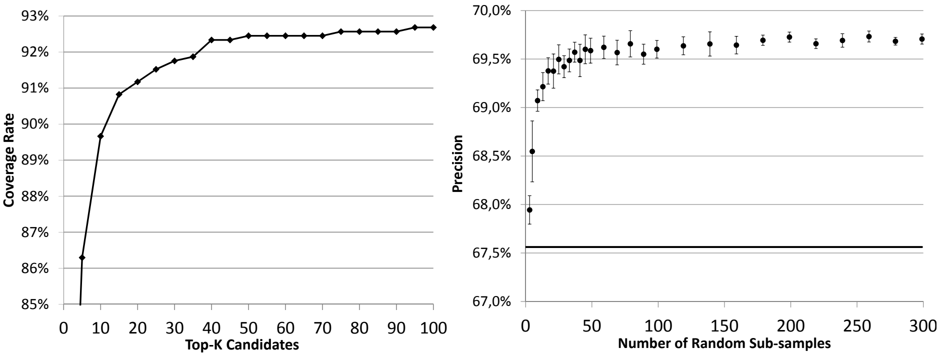
Cucerzan (2007) finds a linking of mentions to Wikipedia entities, such that the sum of vector-based similarities between the candidate entities and the document (containing the mentions) as well as the similarities between pairs of candidate entities is maximized. We refer to this method as LED (Large-scale Entity Disambiguation). The original work has been conducted at Microsoft and the code is proprietary. Hence, we had to re-implement the algorithm according to the descriptions in the paper. To make sure that algorithm was correctly implemented, we evaluated it on the original dataset, and achieved results comparable to those presented in the original paper. Note that, since many entities from Wikipedia are not present in YAGO, the task of linking mentions of the CUCERZAN dataset to YAGO is different from the original task addressed in (Cucerzan, 2007), where mentions were linked to Wikipedia articles.

4.3 Parameter Analysis for BEL

For BEL, the parameters are optimized to deal with common natural-language articles on the Web (e.g., articles from encyclopedic pages or news sites). The same parameter settings are used on all three datasets described above to show the performance of BEL on different types of corpora. To achieve such a common setting of the parameters, we trained BEL on articles sampled from the above datasets, each of which exhibits specific textual characteristics.

4.3.1 Pruning Candidate Lists

In BEL, each mention is assigned a list of candidates. In general, such a list could contain hundreds or even thousands of entities. However, the mention should be linked to at most one entity in the list.



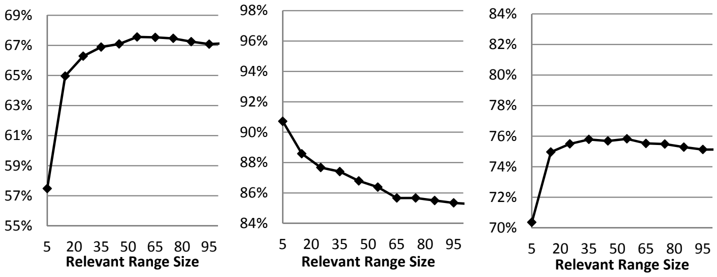
(a) (b)

Figure 2: Parameter analysis experiments (in %): (a) Correct Entity Coverage Rate. (b) Performance of bagging strategy in precision in comparison to the performance of a single classifier.

We randomly picked 1000 mentions from the three datasets to analyze the impact of the candidate list size. The coverage rate (i.e., the relative frequency by which the correct entity is contained in the list) in relation to the list size is shown in Figure 2a. The lists are sorted by decreasing "prominence" scores (see Section 3). In this experiment, 139 mentions have no corresponding entity in YAGO, while 61 correct entities are missing, which means that the maximum coverage rate that a candidate selection strategy can achieve is $800/861 \approx 92.92\%$. As the curve shows, most of the correct entities are indeed located within the top positions of the candidate lists. Therefore, we prune the ranked lists by selecting the top-40 candidates for further processing.

4.3.2 Range of Relevant Terms

As mentioned earlier, the bagging of classifiers is aimed at capturing the contextual information of a mention by randomly sampling terms surrounding it, a process that is repeated several times, once for every ranking classifier. As a sampling procedure we employ bootstrapping (Breiman, 1996), which captures the diversity of contextual information derived from the original range, while mitigating dependencies between terms. We analyze the quality of this bagging strategy mainly based on two criteria: (1) the size of the range of relevant terms, and (2) the bagging size (i.e., number of randomly sampled subsets).



(a) Precision (b) Recall (c) F1-measure

Figure 3: Performance of the language model by increasing the size of the range of relevant terms.

The impact of the range size on the linking quality is intricate, in the sense that a larger range contains more noise, while a smaller one has sparser context. In BEL, the range of relevant terms is empirically calibrated, by evaluating the performance of BEL on different range sizes after removing stop words and non-English terms. To avoid any bias from the "prominence" score and to focus only on the context, we set the $logP(e|m)$-component of the scoring function to 0. Figure 3 shows the average performance based on 10-fold cross validation on all evaluation datasets. The F1-measure achieves maximum when the range of relevant terms contains 55 terms, which is also in agreement with the optimal setting found by Gooi and Allan (2004). Thus, we define the size of the range to be 55. When a document contains less terms, BEL takes the whole text into account.

The bagging strategy of BEL encourages the contextual diversity, while it reduces the linking uncertainty by employing majority voting. Here, to show the impact of our bagging strategy, we randomly pick 80% articles from our datasets for different bagging sizes. Since the bagging strategy is mainly affected by contextual information, we turn off the component responsible for the "prominence" score and run BEL 10 times for each bagging size on these articles. In Figure 2b, the black horizontal line with precision 67.56% is the baseline derived from the single language model classifier. The black dots denote the average precisions and the error bars show the corresponding standard deviation. As the figure shows, by increasing the bagging size the precision increases, while the standard deviation decreases. Considering the precision, efficiency, and stability of the algorithm, we use 199 subsets as the default setting (an odd number of voters is more likely to avoid ties when there are two top-ranked candidates by the voters). Note that, the linking process is stricter and thus leads to a decreased recall. However, the experimental result shows that the impact of the bagging strategy on the increase of precision is consistently higher than its impact on the decrease of the F1-measure; the precision increases from 67.56% to 69.73%, while the F1-measure decreases from 75.82% to 75.32%. Moreover, in our opinion, it is better to suggest that a mention is not in the knowledge base than link it to a wrong entity.

BEL has been evaluated with respect to its linking quality and efficiency. The employed evaluation measures and the results are presented in the following subsections.

4.3.3 Evaluation Measures

For the quality evaluation, we have measured precision, recall, and the F1-measure of each of the above approaches on the mentioned datasets.

A *true positive (tp)* is a mention that has been correctly linked to a YAGO entity. An incorrect linking is defined to be a *false positive (fp)*. Furthermore, a *true negative (tn)* refers to a mention that is correctly identified as an entity that does not occur in YAGO (i.e., non-YAGO entity). The remaining cases are defined as *false negatives (fn)*. Precision is then defined as $P = tp/(tp + fp)$ and recall as $R = tp/(tp + fn)$. The F1-measure is obtained from the harmonic mean of precision and recall as $F = 2PR/(P + R)$.

For the efficiency evaluation, we have measured the runtime (in seconds) of each approach on each dataset.

4.3.4 Evaluation of Linking Quality

The results of the quality evaluation are shown in Table 2, along with the corresponding confidence intervals, which are calculated by repeating 30 times a random sampling of subsets containing 60% of the documents from each dataset. For each dataset, the results computed on all documents are within the intervals that correspond to a confidence level of 99% according to the Student's t-distribution to show that although some of the datasets are of moderate size, the 99% confidence interval of the scores computed on the sampled subsets is relatively small.

As it can be seen, BEL significantly outperforms all the other approaches on the CoNLL-YAGO dataset, especially on precision. Also, for the CUCERZAN dataset, the quality of BEL is comparable to that of AIDA-GRAPH and AIDA-KORE, and it significantly outperforms LED. Moreover, in terms of precision, BEL performs also on this latter dataset significantly better than the other approaches (i.e., from a statistical point of view). Together with BEL's impressive efficiency (see Section 4.3.5), the precision-related quality is a crucial scalability aspect, since when processing a high throughput of documents it is highly important that the produced linkings be rather correct.

For the KORE dataset, AIDA-KORE outperforms other approaches. However, it should be noted that KORE is a very challenging dataset and that the AIDA-KORE approach has been specifically tailored to such datasets. Also note that although the AIDA-KORE algorithm shows a high linking quality in the experiments, it is the least efficient approach, since it performs complex joint reasoning over groups of candidate entities and mentions. In our experiments, we had to wait more than 15 hours for the evaluation results of this approach for KORE dataset, since the mentions contained in this corpus are highly ambiguous.

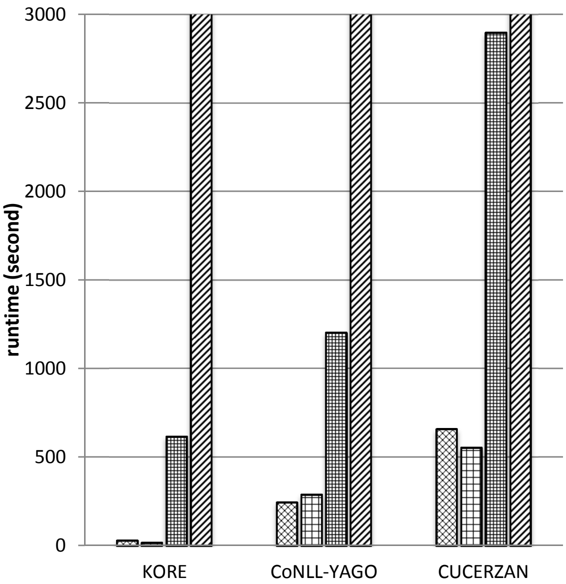
In comparison to a greedy linking strategy, where a mention is simply linked to the most prominent entity according to the "prominence" score, which is our baseline BEL-PROM, BEL performs much better on all three datasets. This fact highlights the importance of the contextual similarity component in the model.

Table 2: Evaluation results (in %).

	Method	Precision	Recall	F1
CoNLL-YAGO	LED	62.35 (-1.92,+0.25)	96.13 (-0.43,+0.27)	75.63 (-1.50,+0.18)
	AIDA-GRAPH	78.67 (-0.80,+1.23)	96.29 (-0.20,+0.64)	86.59 (-0.41,+0.82)
	AIDA-KORE	77.11 (-0.86,+0.80)	96.21 (-0.64,+0.25)	85.61 (-0.67,+0.47)
	BEL-PROM	68.37 (-0.89,+1.30)	97.40 (-0.25,+0.32)	80.30 (-0.61,+0.97)
	BEL	**81.40** (-1.33,+0.78)	95.72 (-0.38,+0.25)	**87.98** (-0.85,+0.46)
CUCERZAN	LED	63.47 (-0.40,+1.01)	96.94 (-0.11,+0.24)	76.72 (-0.28,+0.75)
	AIDA-GRAPH	81.30 (-0.57,+0.16)	94.64 (-0.28,+0.17)	87.47 (-0.40,+0.11)
	AIDA-KORE	81.35 (-0.83,+0.03)	97.31 (-0.25,+0.10)	**88.61** (-0.57,+0.03)
	BEL-PROM	73.92 (-0.53,+0.29)	98.83 (-0.11,+0.06)	84.58 (-0.37,+0.20)
	BEL	**82.37** (-0.31,+0.25)	93.46 (-0.71,+0.27)	87.56 (-0.35,+0.12)
KORE	LED	40.14 (-3.30,+0.88)	100.00 (-0.00,+0.00)	57.28 (-3.52,+0.79)
	AIDA-GRAPH	62.33 (-1.83,+0.93)	100.00 (-0.00,+0.00)	76.79 (-1.43,+0.68)
	AIDA-KORE	**66.67** (-2.29,+1.91)	94.95 (-0.87,+1.82)	**78.33** (-1.60,+1.48)
	BEL-PROM	31.29 (-1.83,+1.47)	100.00 (-0.00,+0.00)	47.67 (-2.23,+1.61)
	BEL	54.55 (-2.40,+2.53)	76.61 (-0.76,+2.20)	63.72 (-1.72,+2.08)

Table 3: Efficiency comparison

Method	KORE	CoNLL-YAGO	CUCERZAN
BEL	30.02s	244.34s	657.44s
LED	17.70s	288.52s	552.26s
AIDA-GRAPH	615.35s	1,202.66s	2,897.43s
AIDA-KORE	>15h	>11h	>25h



4.3.5 Efficiency Evaluation

The lean algorithmic design of BEL enables a highly efficient linking process. For the efficiency comparison, we used a Pentium 3.1GH machine with 8GB of main memory. The indexes for the language models of YAGO2 entities, as well as the indexed YAGO2 knowledge base (that was used by all approaches for the linking) were maintained in a PostgreSQL 9.1 database.

For each dataset, Table 3 shows the runtime of each approach. Obviously, the joint reasoning strategy of AIDA-GRAPH comes at high efficiency costs; on all datasets it has been outperformed by the other approaches. While LED is slightly more efficient than BEL on the KORE and CUCERZAN datasets, as shown in Table 2, it often pays a high cost in terms of quality, and BEL also outperforms it in terms of efficiency on the CoNLL-YAGO dataset. Note that the runtime of LED and BEL are both practically viable from a user's perspective. The AIDA-KORE approach, on the other hand, lacks practical viability, since it needs several hours to process even moderately sized datasets (e.g., approx. 11 hours for the CoNLL-YAGO dataset). For the CUCERZAN dataset, which is the largest one, although the F1-measure of AIDA-KORE is approximately 1% higher than BEL, one needs to wait more than 25 hours to get the result. Instead, BEL can finish the linking process in around 11 minutes.

4.3.6 Discussion

Both, the "prominence" score and the contextual score derived from the proposed bagging strategy have advantages and limitations; they are orthogonal in nature, and their individual strengths are manifested in different ways in the final decision of the algorithm. The value of the "prominence" score has a high impact on the final decision, when BEL is run on articles about famous people, organizations, locations, products, events etc. Typical examples of articles that contain such entities are news reports, scholarly articles containing encyclopedic knowledge, and product descriptions. In contrast, the bagged language models have a high impact on the final decision in cases where the occurring mentions are highly ambiguous but contain valid key information surrounding the mention. Examples for such articles can be found in all three datasets we have used.

Although in many cases, linking the mention to the most prominent candidate entity leads to the correct decision (e.g., "Australia" refers most probably to the country), this strategy is not reliable for many ambiguously used mentions. For example, in one article of CoNLL-YAGO dataset, the named entity "Australia National Cricket Team" in YAGO, was also often referred to by "Australia". Nevertheless, BEL was able to establish a linking to the correct named entity.

The datasets we have annotated and a preliminary online-demo of the algorithm are available online[2].

5 Conclusion

The focus of this work has been on lean and light-weight classification algorithms, which as an ensemble provide a reliable and efficient linking strategy. The comparison of our approach, BEL, with state-of-the-art techniques on manually-labeled, benchmark datasets shows that BEL indeed fulfills the above criteria. Especially on longer, real-world texts, BEL shows an unprecedented quality and efficiency behavior. Further research is needed to understand how such an approach can be optimized for short texts containing highly ambiguous mentions of named entities. We are convinced that BEL provides a robust basis for further research in this area.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. *DBpedia: A nucleus for a web of open data*. The Semantic Web. Springer.

David Aumueller, Hong-Hai Do, Sabine Massmann, and Erhard Rahm. 2005. Schema and ontology matching with COMA++. In *Proceedings of the International Conference on Management of Data*, pages 906–908. ACM.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 79–85.

Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data*, 1(1):5.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the International Conference on Management of Data*, pages 1247–1250.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716.

Fang Du, Yueguo Chen, and Xiaoyong Du. 2013. Linking entities in unstructured texts with RDF knowledge bases. In *Web Technologies and Applications*, pages 240–251. Springer.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2009. Scaling Wikipedia-based named entity disambiguation to arbitrary web text. In *IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 363–370.

[2] `http://hpi-web.de/naumann/projekte/bel.html`

Michael Fleischman and Eduard Hovy. 2004. Multi-document person name resolution. In *Proceedings of the Workshop on Reference Resolution and its Applications*.

Chung Heong Gooi and James Allan. 2004. Cross-document coreference on a large scale corpus. In *Proceedings of the Conference on Natural Language Learning at HLT-NAACL*, pages 9–16.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 2011a. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the International World Wide Web Conference*, pages 229–232.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011b. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. KORE: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 545–554.

Lan Huang, David Milne, Eibe Frank, and Ian H. Witten. 2012. Learning a concept-based document similarity measure. *Journal of the American Society for Information Science and Technology*, 63(8):1593–1608.

Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore Graepel, and Zoubin Ghahramani. 2013. Sigma: Simple greedy matching for aligning large knowledge bases. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 572–580. ACM.

Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1-2):1338–1347, September.

Thomas Lin, Mausam, and Oren Etzioni. 2012. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 84–88.

Gideon S. Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of the Conference on Natural Language Learning at HLT-NAACL*, volume 4, pages 33–40.

David Milne and Ian H. Witten. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25–30. AAAI Press.

Felix Naumann and Melanie Herschel. 2010. *An Introduction to Duplicate Detection*. Morgan and Claypool Publishers.

Ted Pedersen, Amruta Purandare, and Anagha Kulkarni. 2005. Name discrimination by clustering similar contexts. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, volume 3406 of *Lecture Notes in Computer Science*, pages 226–237. Springer.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. LINDEN: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the International World Wide Web Conference*, pages 449–458.

Avirup Sil, Ernest Cronin, Penghai Nie, Yinfei Yang, Ana-Maria Popescu, and Alexander Yates. 2012. Linking named entities to any database. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 116–127.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the International World Wide Web Conference*, pages 697–706. ACM.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the Conference on Natural Language Learning at HLT-NAACL*, volume 4, pages 142–147.

Melanie Weis and Felix Naumann. 2005. DogmatiX tracks down duplicates in XML. In *Proceedings of the International Conference on Management of Data*, pages 431–442, Baltimore, MD.

Fei Wu and Daniel S. Weld. 2008. Automatically refining the wikipedia infobox ontology. In *Proceedings of the International World Wide Web Conference*, pages 635–644.

Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, April.