Robust Named Entity Disambiguation with Random Walks

Editor(s): Harith Alani, Open University, UK
Solicited review(s): Ricardo Usbeck, Universität Leipzig, Germany; Mena B. Habib, Maastricht University, Netherlands; Ziqi Zhang, Nottingham Trent University, UK

Zhaochen Guo and Denilson Barbosa [*]
Department of Computing Science, University of Alberta, Edmonton, AB, Canada.
E-mail: {zhaochen, denilson}@ualberta.ca

Abstract. Named Entity Disambiguation is the task of assigning entities from a Knowledge Graph (KG) to *mentions* of such entities in a textual document. The state-of-the-art for this task balances two disparate sources of similarity: lexical, defined as the pairwise similarity between mentions in the text and names of entities in the KG; and semantic, defined through some graph-theoretic property of a subgraph of the KG induced by the choice of entities for each mention. Departing from previous work, our notion of semantic similarity is rooted in Information Theory and is defined as the mutual information between random walks on the disambiguation graph induced by choice of entities for each mention. We describe an iterative algorithm based on this idea, and show an extension that uses learning-to-rank, which yields further improvements. Our experimental evaluation demonstrates that this approach is robust and very competitive on well-known existing benchmarks. We also justify the need for new and more difficult benchmarks, and provide an extensive experimental comparison of our method and previous work on these new benchmarks.

Keywords: Named entities, entity linking, entity disambiguation, relatedness measure, random walk, benchmarking.

1. Introduction

A knowledge graph (KG) is a repository of structured information consisting of unique entities (e.g., notable people, cities, companies and other kinds of organizations, etc.), facts about entities (e.g., the date of birth of such people), and relations between entities (e.g., the cities where such people were born). The recent advent of large KGs, derived from Web-scale corpora and/or Wikipedia, has renewed the interest in algorithmic understanding of natural language text, especially in the context of the Web and social media where facts or properties about named entities are described in many documents.

Two crucial tasks in natural language understanding have to do with *named entities*, which are the persons, organizations, locations, etc. that are explicitly mentioned in text using proper nouns: (1) *Named Entity Recognition* (NER) corresponds to finding *mentions* to entities in the text; and (2) *Named Entity Disambiguation* (NED), which is the task of disambiguating the named entities by linking them to the actual entities in the KG (when possible). The NER process is usually done by taking lexical and grammatical features into account, meaning that some of the mentions identified through this process may refer to entities that are not in the KG. NED, on the other hand, is done for a specific KG, and provides a mapping between each mention and an existing KG entity (or NIL if no such assignment is possible), thus grounding each mention in a surrogate to a real world entity. NED is key for the KG construction and maintenance which can expand or correct KGs with (new) facts of entities extracted from previously unseen text [17].

Our work concerns the NED task. This article describes effective algorithms for solving this problem,

[*] Corresponding author, e-mail: denilson@ualberta.ca

1570-0844/0-1900/$27.50 © 0 – IOS Press and the authors. All rights reserved

assuming the input is a KG and a document where all mentions to entities (explicit or implicit) have been identified.

Challenges. The ambiguity of natural language has always been a challenge for NED, even to humans, because most real world entities can be referred to in many different ways (e.g., people have nicknames), while the same textual mention may refer to multiple real-world entities (e.g., different people have the same name). The following examples illustrate the issues:

Example 1
Malone, a retired professional basketball player, is mostly known for his time with the Washington Bullets, where he was an NBA All-Star twice. He also played for Utah, Philadelphia, and Miami.

Example 2
Malone, nicknamed "The MailMan" spent his first 18 seasons in NBA with the Utah Jazz, and final season with Los Angeles. He was a two-time NBA MVP, a 14-time NBA All-Star

Observe that the same entity (the NBA franchise team *Utah Jazz*) is referred to in different ways in the examples above: as "Utah" in Example 1 and explicitly by its full name in Example 2. On the other hand, two different players are mentioned in the same way, by their last name "Malone": *Jeff Malone* in Example 1 and *Karl Malone* in Example 2. The disambiguation of the mentions to the players is hard because of the shared context: both mentions refer to *basketball* players who played as an *NBA All-Star*, and also played for the same franchise.

1.1. Canonical Solution

A typical NED system proceeds in two stages: (1) *candidate selection*, which aims at quickly finding a small number of KG objects which are likely to be mentioned in the text, and (2) *mention disambiguation* which computes the final mapping between the named entities in the text and the objects in the KG. Candidate selection is often started by consulting alias dictionaries (see, e.g., [35]) using coarse-grained string similarity matching to minimize the chances of filtering out good candidates. The disambiguation step, on the other hand, is done by aggregating multiple and much more sophisticated notions of similarity.

The first disambiguation methods assumed that the mentions in the text were completely independent, and relied on *local* contextual features such as the words surrounding the mentions [2,3], and *statistical* features derived from KGs. These approaches work best when the surrounding context is rich enough to uniquely identify the objects being mentioned. For instance, these methods would work really well with famous politicians with an uncommon last name. On the other hand, they would fail to distinguish the two NBA players in the examples above given their shared context (from a lexical point of view): both players have the same last name and played for the same team. Indeed, "Malone" in both examples is likely to be mapped to *Karl Malone*, who is more well known and thus has a higher prior.

To avoid being fooled by disproportional priors, most state-of-the-art approaches exploit known connections between the objects in the KG to help with the disambiguation, based on the assumption that the disambiguation of each mention should somehow affect the disambiguation of another. For instance, Example 1 has an explicit mention to the NBA team *Washington Bullets*, which is easy to disambiguate; once that is done, the NED system must be significantly more likely to map "Malone" in that sentence to *Jeff Malone*, since he is more strongly related to that team. These associations between objects in the KG induce notions of semantic relatedness which are often captured as some property of a *disambiguation graph* containing objects in the KG which were identified as *candidates* for the mentions in the text as illustrated in Fig. 1.

The choice of semantic similarity determines both the accuracy and the computation cost of the NED method. One successful strategy [15] computes the set-similarity involving (multi-word) *keyphrases* about the mentions and the entities, collected from the KG. This approach works best when the named entities in the document are mentioned in similar ways to those in the corpus from which the KG is built (typically, Wikipedia). Another approach [27] computes the set similarity between the neighbor entities directly connected in the KG. Doing so, however, ignores entities that are indirectly connected yet semantically related, making limited use of the KG graph.

In previous work [11], we introduced a method that used an information-theoretic notion of similarity based on stationary probability distributions resulting from random walks [38] on the disambiguation graph, which led to consistently superior accuracy. This article describes substantial extensions over that method.

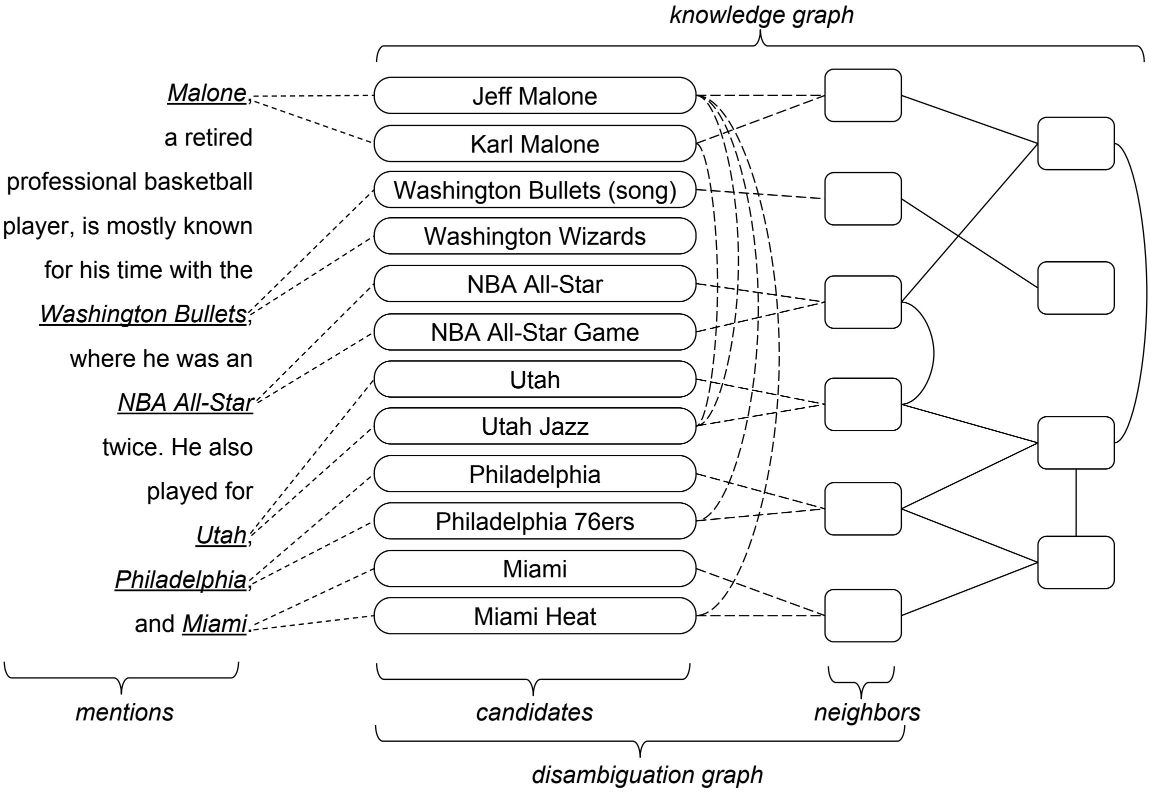

knowledge graph
Malone, a retired professional basketball player, is mostly known for his time with the Washington Bullets, where he was an NBA All-Star twice. He also played for Utah, Philadelphia, and Miami.
Jeff Malone
Karl Malone
Washington Bullets (song)
Washington Wizards
NBA All-Star
NBA All-Star Game
Utah
Utah Jazz
Philadelphia
Philadelphia 76ers
Miami
Miami Heat
mentions
candidates
neighbors
disambiguation graph

Fig. 1. Example named entity disambiguation scenario.

1.2. Our approach

Our WNED (Walking Named Entity Disambiguation) method is a greedy, global NED algorithm based on a sound information-theoretic notion of semantic relatedness derived from random walks on carefully built disambiguation graphs [11]. We build specific disambiguation graphs for each document, thus adhering to the notion of global coherence assumption—that coherent entities form a dense subgraph. By virtue of using random walks, our notion of similarity leverages indirect connections between nodes in the disambiguation graph, and is thus less susceptible to false positives incurred by disproportionately high priors of head entities.

Contributions This article presents several significant extensions over our previous work:

- a revised iterative disambiguation algorithm with fewer parameters and a simplified optimization goal than in [11];
- a new disambiguation method based on a state-of-the-art "learning-to-rank" approach that further improves on the accuracy reported previously;
- a comparative evaluation of both approaches against 11 other NED systems on 16 public datasets using a the GERBIL framework [40];
- a much deeper experimental evaluation than previous works in the area, establishing that previous public benchmarks are rather "easy", in the sense that a simple baseline can correctly disambiguate most mentions;
- a framework for deriving new benchmarks and two benchmarks from large Web corpora (Wikipedia and Clueweb 2012) with documents of increasing difficulty, which are also *balanced* (i.e., they have the same number of documents in each difficulty class).

We evaluate the learning-to-rank approach in two ways. First, we follow the the standard machine learning methodology (namely to separate the benchmarking data into training and testing sets) for each dataset. Next, we use the widely used CoNLL dataset for training, and test the resulting system on all other benchmarks. This results in an algorithm that consistently outperforms previous methods, across benchmarks and often by a wide margin.

It is also worth noting that our statically hand-tuned algorithm also outperformed most of previous methods and is quite competitive with the learning approach. These observations corroborate the superiority and robustness of using random walks for the NED task. In summary, the algorithm and the evaluation methodology described in this article significantly push the state-of-the-art in this task.

2. NED with Random Walks

This section first describes Named Entity Disambiguation as an optimization problem and then gives an overview of our solution based on using Random Walks to estimate semantic similarities and a greedy, iterative approximation algorithm (Section 3).

2.1. The NED Problem

Let d be a document with all mentions to named entities marked up through an NER process, and $KG = (E, L)$ be a knowledge graph represented as a graph whose nodes in E correspond to real world entities and links in L capture relationships among them. The task of NED is to assign unique entity identifiers from E to the mentions in d, whenever appropriate. NIL represents the entities that do not exist in the KG (also known as *out-of-KG* entity).

More precisely:

Definition 1 (Named Entity Disambiguation) *Given a set of mentions $M = \{m_1, \ldots, m_m\}$ in a document d, and a knowledge graph $KG = (E, L)$, the NED problem is to find an assignment $\Gamma : M \rightarrow E \cup \{\mathsf{NIL}\}$.*

A good assignment Γ balances two factors: the *local* similarity between mention m_i and the entity $e_j = \Gamma(m_i)$ assigned to it, and the *global* coherence among the entities in the assignment.

As usual, we define the local similarity $\phi(m_i, e_j)$ as:

$$\phi(m_i, e_j) = \alpha\, prior(m_i, e_j) + (1-\alpha)\, ctx(m_i, e_j) \quad (1)$$

where $prior(m_i, e_j)$ is a corpus prior probability that e_j is the right entity for m_i, usually derived from alias dictionaries built from the KG, and $ctx(m_i, e_j)$ is the similarity between *local* features extracted from text (e.g., keywords) surrounding m_i in the document and descriptions associated to e_j in the KG.

The global coherence $\Psi(\Gamma)$ of the assignment measures how each entity in the assignment relates to the others:

$$\Psi(\Gamma) = \sum_{e \in \Gamma[M]} \psi(e, \Gamma) \quad (2)$$

in which $\psi(e, \Gamma)$ measures the semantic similarity between an entity e and all others in the assignment Γ. Maximizing the sum in Eq. 2 is consistent with the *document coherence assumption*, in which one expects the input document to belong to a single *topic* (e.g., sports) under which all entities in the assignment Γ are tightly related.

Under the reasonable assumption that the local similarity is normalized, we can formulate the NED problem as a min-max optimization where the goal is to *maximize* the global coherence while minimizing the *loss* in pairwise local similarity within the assignment, which can be estimated as $|M| - \sum_{m_i \in M} \phi(m_i, \Gamma(m_i))$. Here $|M|$ is the number of mentions in the document. An equivalent and simpler formulation of the problem is to find an assignment Γ^* that maximizes:

$$\Gamma^* = \arg\max_{\Gamma} \left(\Psi(\Gamma) \cdot \sum_{m_i, e_j \in \Gamma} \phi(m_i, e_j) \right) \quad (3)$$

The primary role of $\Psi(\Gamma)$ in the optimization above is to leverage connections between entities in the KB to prevent disambiguation mistakes caused by disproportionately high priors of some candidate entities. For example, *Karl Malone* has a higher prior than *Jeff Malone* and thus would be incorrectly assigned to mention "Malone" in the sentence of Example 1 above. However, once "Washing Bullets" is disambiguated, $\Psi(\Gamma)$ will "counter" the effect of the high prior because *Jeff Malone* is directly connected to that team in KB.

The state-of-the-art is to define $\Psi(\Gamma)$ as some graph-theoretic measure derived from a subgraph of the KB induced by the assignment (recall Fig. 1). For example, the AIDA system assigns weights to edges in the (disambiguation) subgraph and takes the sum of the weights of those edges in a minimum spanning tree connecting all entities used in the assignment. Thus, if e_1 and e_2 are two candidate entities for the same mention m, AIDA will favor the one with the shortest path (in terms of KB edges) to another entity also in the assignment.

2.2. Global Coherence with Random Walks

Our approach to capturing the global coherence [11] is rooted in Information Theory and corresponds to the mutual information between probability distributions arising from *random walk* processes on the disambiguation graph: one always restarting from a single candidate entity, and the other restarting from *all entities* used in the assignment Γ.

More precisely, we build a disambiguation graph G (Fig. 1) which is a subset of the KB: the nodes are candidate entities and their immediate neighbors, and the edges are the associations between these entities in the KB. Let N be the number of nodes in G. A random walk with restart is an iterative process that assigns scores to nodes in the graph, defined as:

$$r^{t+1} = \beta \times r^t \times A + (1 - \beta) \times \mathbf{v} \tag{4}$$

where r is the N-dimensional vector with the scores, A is the transition matrix for the graph, $\mathbf{v}$ is the *preference vector*, used to determine the nodes in the graph from which new walks (re)start, and β is the probability of following an edge in the graph.

Given any entity e in the graph, if we define the preference vector as $\mathbf{v}_i = \mathbb{1}(index(e) = i)$, the corresponding random walk on G will induce a probability distribution of reaching any of the N nodes in G starting from e. We call such distribution the *semantic signature* of the entity e. This notion of signature extends naturally to a set of entities; thus, the signature of an assignment Γ is the probability distribution resulting from a random walk process where the starting points are the entities in Γ.

We define $\psi(e, \Gamma)$ in Eq. 2 as the reciprocal of the *mutual information* between the signatures of e and Γ, measured using a variant of the Kullback–Leibler divergence to handle the case when Q_i is zero:

$$ZKL_\gamma(P, Q) = \sum_i P_i \begin{cases} \log \frac{P_i}{Q_i} & Q_i \neq 0 \\ \gamma & Q_i = 0 \end{cases} \tag{5}$$

Intuitively, our notion of coherence favors candidate entity e_1 over e_2 if its signature is more similar to the signature induced by the assignment Γ. By leveraging random walks, our notion of coherence takes into account both direct and indirect connections between pairs of entities in the disambiguation graph, departing from previous work.

Algorithm 1 Iterative WNED

Input: $M = \{m_1, m_2, \ldots, m_n\}, KG = (E, L)$
Output: Assignment $\hat{\Gamma} : M \rightarrow E \cup \{\mathsf{NIL}\}$

```
$\hat{\Gamma} = \langle \Gamma_i, 1 \leq i \leq |M| \mid \Gamma_i = \mathsf{NIL} \rangle$
$L = \langle m_i \in M$ sorted by $|aliases(m_i)| \rangle$
for $i = 1$ to $|L|$ do
    if $|cand(m_i)| = 1$ then
        $\hat{\Gamma}_i(m_i) = cand(m_i)$
    end if
    if $|cand(m_i)| > 1$ then
        $\mathbf{d}$ = vecInit($M, KG, \hat{\Gamma}$); $Q = signature(\mathbf{d})$
        $max = 0$
        for $e_j \in cand(m_i)$ do
            $P = signature(e_j)$
            $\psi(e_j, \hat{\Gamma}_{i-1}) = \frac{1}{ZKL_\gamma(P, Q)}$
            $score(e_j) = \psi(e_j, \hat{\Gamma}_{i-1}) \cdot \phi(m_i, e_j)$
            if $score(e_j) > max$ then
                $e^* = e_j$ ; $max = score(e_j)$
            end if
        end for
        if $score(e^*) < \theta$ then
            $\hat{\Gamma}_i(m_i) = \mathsf{NIL}$
        else
            $\hat{\Gamma}_i(m_i) = e^*$
        end if
    end if
end for
return $\hat{\Gamma}$
```

2.3. Linking to NIL

A mention is linked to NIL when no good candidate entities can be found for that mention or when the similarity score of the best entity falls short of a threshold. Both criteria are, of course, application-specific, and thus outside of the scope of this work.

3. Disambiguation Algorithms

This section gives the details of our two NED methods, which use the same underlying algorithm and differ only on the way they compute the semantic similarity.

As previously observed (see, e.g., [14]), the NED problem is intimately connected with a number of NP-hard optimizations on graphs, including the maximum m-clique problem [10], from which a polynomial time reduction is not hard to construct. Thus we resort to

Algorithm 2 vecInit

```
Input: $M = \{m_1, m_2, \ldots, m_n\}, KG = (E, L), \Gamma : M \to E$
Output: Document disambiguation vector $\mathbf{d}$
let $n$ be the size of the disambiguation graph
$\mathbf{d} = \mathbf{0}_{(n)}$
if $\Gamma \neq \emptyset$ then
    for $m, e \in \Gamma$ do
        $\mathbf{d}_e = 1$
    end for
else
    for $m \in M$ do
        for $e \in cand(m)$ do
            $\mathbf{d}_e = prior(e, m) \cdot tfidf(m)$
        end for
    end for
end if
normalize $\mathbf{d}$
return $\mathbf{d}$
```

an iterative greedy algorithm, called Walking NED (WNED), and described in Alg. 1.

WNED starts by sorting the mentions by their degree of ambiguity, measured by the number of entities that have that mention as an alias (line 2). Note that the ambiguity of a mention is typically much higher than the number of candidates that are in fact considered (see [11]). If a mention has a single promising candidate, WNED assigns that candidate to the mention (line 5). The main loop of the algorithm goes through each mention with more than one promising candidate (lines 7–23): updating the semantic signature of the partial entity assignment (line 8), and, for each candidate, computing the signature of the candidate (line 11), and selecting the best candidate based on the following greedy approximation of the original optimization:

$$\hat{\Gamma}_i(m_i) = \underset{e_j \in cand(m_i)}{\arg\max} \left(\psi(e_j, \hat{\Gamma}_{i-1}) \cdot \phi(m_i, e_j)\right) \tag{6}$$

A final step of the algorithm is to assign NIL to those mentions whose even the best candidate entity has a low score (lines 18–22). The cut-off threshold θ is application-defined.

Parameters The experimental evaluation reported here was obtained with the following parameter setting: in Eq. 1 $\alpha = 0.8$; in Eq. 4, $\beta = 0.85$; and in Eq. 5, $\gamma = 20$. These settings were obtained experimentally.

3.1. Disambiguation via Learning to Rank

Most approaches including ours consider the NED problem as an entity ranking problem: candidates are ranked according to a score (e.g. E.q 6), and the highest ranked candidate is assigned. Such ranking is based on multiple criteria (e.g., prior probability, context similarity, semantic similarity, etc.) that may apply differently in different situations, making it hard or impossible to craft a fixed way of aggregating them all that performs well in all cases. With benchmarking data available, one can leverage machine learning to derive a better ranking strategy, at least for the specific dataset used for training.

This Learning to Rank approach originates from Information Retrieval. The methods can be divided into three classes [21]: *pointwise*, *listwise*, and *pairwise*. The *pointwise* approaches consider query-document pairs as independent instances, and use regression to predict the scores of each document (given the query) and rank them accordingly. As such, *pointwise* methods are not trained on actual rankings, but instead on features from the documents and the queries. On the other hand, *listwise* approaches are trained on the actual document rankings for different queries, and their goal is to learn how to predict a new ranking given a new query. Finally, the *pairwise* approaches work on ordered pairs of documents instead of full rankings, and seek to predict which document in the pair would rank higher.

Among the Learning-to-Rank approaches we experimented with, LambdaMART [41], which is a *pairwise* method, achieved the best performance. LambdaMART employs the MART (Multiple Additive Regression Trees) algorithm to learn Boosted Regression Trees, and takes advantage of LambdaRank [4] gradients to bypass the difficulty of handling non-smooth cost function introduced by most IR measures, and combines the cost function and IR metrics together to utilize the global ranking structure of documents.

Our Learning-to-Rank WNED algorithm, called L2R.WNED, is essentially the same as Alg. 1, except we replace the scoring of entities given the mentions (lines 8–13) by invoking the LambdaMART classifier to obtain the highest ranked candidate entity. The details follow.

Features Although there are many useful features for ranking [45], our main goal is to establish the robustness and utility of the semantic similarity for the NED task, rather than performing exhaustive feature engineering at the risk of over-fitting. Thus we use four features, all of which are familiar in this research area: prior probability, context similarity, semantic relatedness, and *name similarity* which is measured by the N-Gram distance [19] between a mention and the canonical name of the entity. More precisely, given a mention-entity pair $m-e$, we extract the following features: (1) prior probability $prior(m, e)$, (2) context similarity $ctx(m, e)$, (3) name similarity $nameSim(m, e)$, and (4) semantic relatedness $semantic(e, \mathbf{d})$ as before.

Training data Using a *pairwise* ranking method, for each mention we need *ordered* pairs of entities e_1, e_2 such that e_1 is ranked *higher* than e_2. Such pairs are easy to obtain when gold standard benchmarking data is available. Given a mention m, let entity $\tilde{e}$ be the one assigned to m in the ground truth. We apply the candidate selection step (see [11]), collecting k candidate entities $c_1, \ldots, c_k$ for the mention. Our training data is then all pairs $(\tilde{e}, c_i), \tilde{e} \neq c_i, 1 \leq i \leq k$.

In our experiments, we verify the system performance with two training approaches. First, we used the standard 10-fold cross-validation, thus training and testing on the same benchmark. Also, we experimented with training on the CoNLL dataset, which is the largest of the public benchmarks, while testing on other datasets. Both approaches worked well. In particular, we found that training CoNLL data produced a quite robust method that worked well on other benchmarks.

Prediction Using the training data, our approach trains a model and uses it to build an evaluator which takes in a feature set representing the similarity between a mention and its candidate entity and computes the probability that the mention refers to the candidate. All candidate entities are evaluated and ranked with the probability. The entity with the highest probability is then chosen as the final entity. More specifically, given the feature set $Feature(m_i, e_j)$: $prior(m_i, e_j)$, $ctx(m_i, e_j)$, $nameSim(m_i, e_j)$, $semantic(e, \mathbf{d})$, the evaluator aims to find the entity maximizing the following objective.

$$e^* = \underset{e_j \in cand(m_i)}{\arg\max} \; evaluate(Feature(m_i, e_j)) \quad (7)$$

3.2. Computational Cost

There are two factors contributing to the cost of WNED: computing the signatures and greedily selecting the best candidate for each mention.

Let $n = |M|$. The total number of candidates that are considered by WNED is $K = O(n)$, because of the candidate selection step keeps a constant number of promising candidates [11]. The total number of semantic signature computations is $K + |M| = O(n)$. The size of the disambiguation graph is $O(n)$ vertices and $O(n^2)$ edges (unless some non-trivial pruning is performed). Therefore, if the number of iterations in the random walks is fixed, computing all signatures can be done in $O(n^2)$ time (and space).

As for the time required for the actual scoring, for fixed KG and input document, computing the prior probability and the context similarity are done through database lookups at $O(1)$ time. In the standard WNED, we also need to compute the Zero-KL divergence on vectors of length n, which can be done in $O(n)$ time. For L2R.WNED, the time is bound by the size of the model, which depends on the amount of training data, and is again $O(1)$ for the purposes of our analysis.

In our experience, the highest actual costs lie in building the disambiguation graphs, which must be done for each input document, and performing the random walks. Our current implementation keeps the entire disambiguation graph in main memory to speed this up. We leave for future work improving the performance of WNED.

4. Experimental Validation

Given the host of applications where NED is useful and the inherent difficulty of the problem, a lot of effort has been devoted recently in establishing fair and comprehensive benchmarks for this task. In particular, Web-based experimental platforms such as GERBIL [40] are a clear step in the right direction, as they go a long way in automating the collection and reporting of results of different algorithms under the same benchmarks and evaluation conditions. In the appendix we report the results of both our methods on GERBIL version 1.2.4 with the D2KB setting.

Despite their effectiveness and convenience, the information reported by platforms such as GERBIL (particularly, aggregate accuracy measurements) is not enough for a deeper analysis that can lead to algorithmic improvements. Thus, we re-implemented and experimented with several NED systems and compared them against both WNED and the L2R.WNED. In the remainder of this section we report on our own experimental evaluation on well-known publicly

Table 1

Accuracy results of all methods on the 4 public benchmarks.

Method	MSNBC			AQUAINT			ACE2004			AIDA-CoNLL		
	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA
PRIOR	0.86	0.86	0.87	0.84	0.87	0.87	**0.85**	0.85	0.87	0.75	0.75	0.76
CONTEXT	0.77	0.78	0.72	0.66	0.68	0.68	0.61	0.62	0.57	0.40	0.40	0.35
Cucerzan	0.88	0.88	0.88	0.77	0.79	0.78	0.79	0.79	0.78	0.73	0.74	0.72
M&W	0.68	0.78	0.80	0.80	0.85	0.85	0.75	0.81	0.84	0.60	0.68	0.68
Han11	0.88	0.88	0.88	0.77	0.79	0.79	0.72	0.73	0.67	0.62	0.62	0.58
AIDA	0.77	0.79	0.76	0.53	0.56	0.56	0.77	0.80	0.84	0.78	0.79	0.79
GLOW	0.66	0.75	0.77	0.76	0.83	0.83	0.75	0.82	0.83	0.68	0.76	0.71
RI	0.89	0.90	0.90	0.85	0.88	0.88	0.82	0.87	0.87	0.79	0.81	0.80
WNED	0.89	0.90	0.90	**0.88**	**0.90**	**0.90**	0.83	0.86	0.89	0.84	0.84	0.83
L2R.WNED-CoNLL	**0.91**	**0.92**	**0.92**	0.85	0.87	0.87	**0.85**	**0.88**	**0.90**	**0.89**	**0.89**	**0.89**
L2R.WNED	**0.91**	**0.92**	0.91	**0.88**	**0.90**	**0.90**	**0.85**	**0.88**	0.89			

available benchmarks. Further, we justify the need for more challenging benchmarks for this task, provide a methodology for deriving such benchmarks, and report on experiments on two new benchmarks we introduce.

Experiment Configuration We refer to previous work [11] for the details on building the entity graph and the initial pruning of candidate entities prior to the actual disambiguation per se. The KB used in our experiment is built from the Wikipedia 20130606 dump. The source code for our system is available from https://github.com/U-Alberta/wned.

Metrics We use the standard *accuracy, precision, recall*, and *F1*:

$$accuracy = \frac{|truth \cap result|}{|truth \cup result|}$$

$$precision = \frac{|truth \cap result|}{|result|}$$

$$recall = \frac{|truth \cap result|}{|truth|}$$

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

Where $truth$ is a ground truth assignment and $result$ is the assignment produced by the NED system.

4.1. Evaluation on Established Benchmarks

We compare WNED and L2R.WNED to the state-of-the-art systems (Detailed descriptions of these systems are in the Related Work at Section 5):

- Cucerzan [7]—the first global NED approach,
- M&W [26]—a leading machine learning NED solution,
- Han11 [13]—a global method that also uses random walks (on a disambiguation graph built differently from ours),
- AIDA [14]—a global method that formulates NED as a subgraph optimization problem,
- GLOW [31]—a system combining local and global features for NED,
- RI [6]—the start-of-the-art NED system using relational inference for mention disambiguation.

We also evaluate two useful baselines: CONTEXT which chooses the candidate entity with highest textual similarity to the mention, $ctx(m, e)$, and PRIOR which picks the entity with highest prior probability for each mention, $prior(m, e)$. These baselines are informative as virtually all methods rely on these measures in one way or another, including ours (recall Eq. 6). Somewhat surprisingly, as shown next, not every method improves on both of them.

As mentioned in [9], GERBIL uses an old version of the public datasets. Thus we update four widely used public benchmarks: (1) MSNBC [7], with 20 news articles from 10 different topics (two articles per topic) and 656 linkable mentions in total; (2) AQUAINT, compiled by Milne and Witten [26], with 50 documents and 727 linkable mentions from a news corpus from the Xinhua News Service, the New York Times, and the Associated Press; (3) ACE2004 [31], a subset of the documents used in the ACE2004 Coreference documents with 35 articles and 257 linkable mentions, annotated through *crowdsourcing*; and (4) AIDA-CoNLL [14], a hand-annotated dataset based on the CoNLL 2003 data, with 1388[1] Reuters news articles and 27817 linkable mentions.

To avoid discrepancy of results due to different Wikipedia versions used in different systems, we update all datasets and results of compared NED systems to their redirected entities in our Wikipedia dump. All datasets used in this evaluation, including the new benchmarks introduced below, as well as the accuracy results obtained with each method on each document can be downloaded from http://dx.doi.org/10.7939/DVN/10968.

Table 1 shows the results of the two baselines and the above listed NED systems on the four public benchmarks. As customary, we report F1 aggregated across mentions (micro-averaged, indicated as **F1@MI**) and across documents (macro-averaged, **F1@MA**).

For the learning to rank approaches, L2R.WNED-CoNLL refers to the method where the learning is done on the AIDA-CoNLL dataset, regardless of the test corpus, and L2R.WNED is the method where the model is trained on a fraction of the respective benchmark. Note that our WNED uses a different optimization objective, thus the results are not the same as the results in our previous work [11].

Discussion A few observations are worth making here. Among previous work, RI has the best performance across benchmarks. The disambiguation via textual similarity alone, as done by the Context baseline, leads to poor accuracy in general, especially on the more challenging AIDA-CoNLL benchmark. The Prior baseline, on the other hand, performs well across the board, outperforming several systems. This points to limitations in the benchmarks themselves: they use high quality news articles, where the entities are likely to be mentioned at least once by their full name (which is easy to disambiguate with a prior alone).

The reader will notice that virtually every method in the literature is evaluated against a baseline like Prior, and if one looks back to earlier works, the reported accuracy of such baseline is not nearly as high as what we report. This can be explained by the continuous cleaning process on Wikipedia—from which the statistics are derived. As we use a more recent and cleaner corpus, where the support for good and appropriate entity aliases is markedly higher than for spurious or inappropriate mentions.

With respect to WNED and L2R.WNED, both outperform all competitors on all benchmarks, with L2R.WNED performing best overall. Another observation is that training our L2R.WNED with AIDA-CoNLL data is quite effective on *all* other benchmarks, and sometimes superior to training our method with data from the specific benchmark. While not surprising (as all benchmarks come from the same domain—news), these results mean that L2R.WNED trained on AIDA-CoNLL can be seen as an effective and off-the-shelf NED system. Another general observation is that there is quite a lot of variability in the relative ordering of the previous methods across benchmarks, except for RI and our methods. This somewhat surprising lack of robustness in some systems may have been caused by over-tuning for the development benchmark, resulting in poor generalization when tested on different benchmarks.

4.2. The Need for New Benchmarks

Although the four benchmarks discussed above are useful reference points, since they are well-known and have been used for the evaluation of most NED systems, they leave a lot to be desired for a deeper and more systematic accuracy evaluation. As noted in the previous section, they are clearly biased towards popular entities, and thus, not representative of all scenarios where NED is necessary. To further illustrate the point, Table 2 breaks down the number of documents in each benchmark at different levels of accuracy achieved by Prior (i.e., the brackets are determined by the overall accuracy of all mentions in the document). As can be seen, the vast majority of documents in all previous benchmarks are not particularly challenging: In fact, Prior produces perfect results for as many as 20% of all documents of AQUAINT and AIDA-CoNLL and 31% of all documents in the case of the ACE2004 benchmark. It follows that these benchmarks are dated and unlikely to lead to further significant improvements in the area.

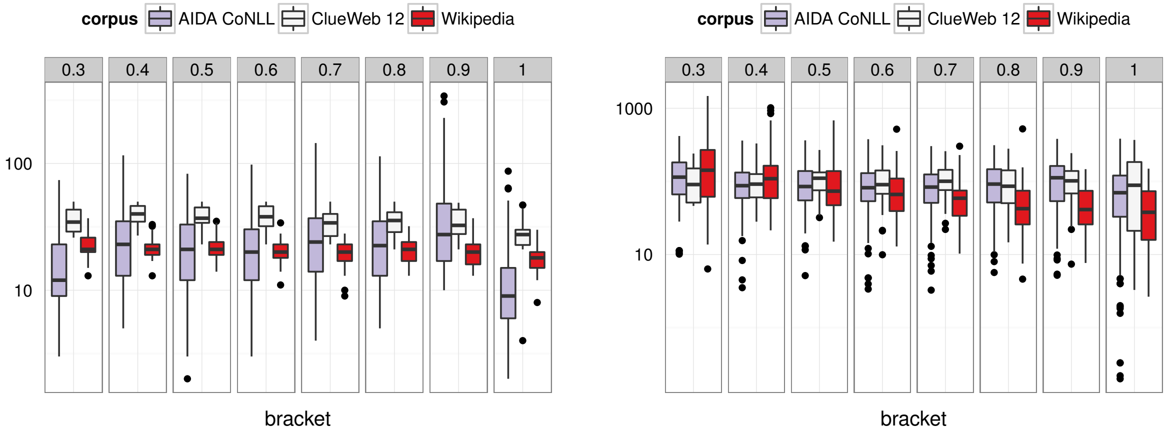
A desirable feature of any thorough evaluation that is not necessarily fulfilled by any of the previous benchmarks is that of *representativeness*. Namely, it would be ideal to have a mix of mentions or documents with *different levels of difficulty* in equal proportions (say on a 10-point scale from “easy” to “hard”). Without such equity, the effectiveness metrics reported in the literature (which aggregate at the mention or doc-

[1] The original dataset includes 5 other documents where all mentions are linked to NIL, and are therefore removed from our analysis.

Table 2

Breakdown of the public benchmarks by the accuracy of the PRIOR method; #docs and #mentions are, respectively, the number of documents and the average number of mentions per document in each bracket; the number in parenthesis is the fraction of the entire benchmark covered by each bracket.

Accuracy	MSNBC		AQUAINT		ACE2004		AIDA-CoNLL	
	#docs	#mentions	#docs	#mentions	#docs	#mentions	#docs	#mentions
0.0 – 0.1	0 (0%)	0	0 (0%)	0	0 (0%)	0	5 (0.4%)	5.0
0.1 – 0.2	0 (0%)	0	0 (0%)	0	0 (0%)	0	35 (2.5%)	40.4
0.2 – 0.3	0 (0%)	0	0 (0%)	0	0 (0%)	0	29 (2.1%)	20.2
0.3 – 0.4	0 (0%)	0	0 (0%)	0	0 (0%)	0	62 (4.5%)	17.4
0.4 – 0.5	2 (10%)	51.5	0 (0%)	0	0 (0%)	0	61 (4.4%)	30.0
0.5 – 0.6	3 (15%)	45.7	0 (0%)	0	0 (0%)	0	100 (7.2%)	22.5
0.6 – 0.7	3 (15%)	37.0	1 (2%)	8.0	5 (14.3%)	10.8	164 (11.8%)	21.7
0.7 – 0.8	4 (20%)	29.8	12 (24%)	15.3	5 (14.3%)	10.8	210 (15.1%)	26.8
0.8 – 0.9	3 (15%)	53.0	16 (32%)	14.4	12 (34.3%)	8.5	267 (19.2%)	28.3
0.9 – 1.0	3 (15%)	25.0	11 (22%)	15.0	2 (5.7%)	12.0	164 (11.8%)	43.5
1.0	2 (10%)	17.5	10 (20%)	13.9	11 (31.4%)	6.4	291 (21.0%)	13.2



(a) Mentions per document.

(b) Candidate entities per mention.

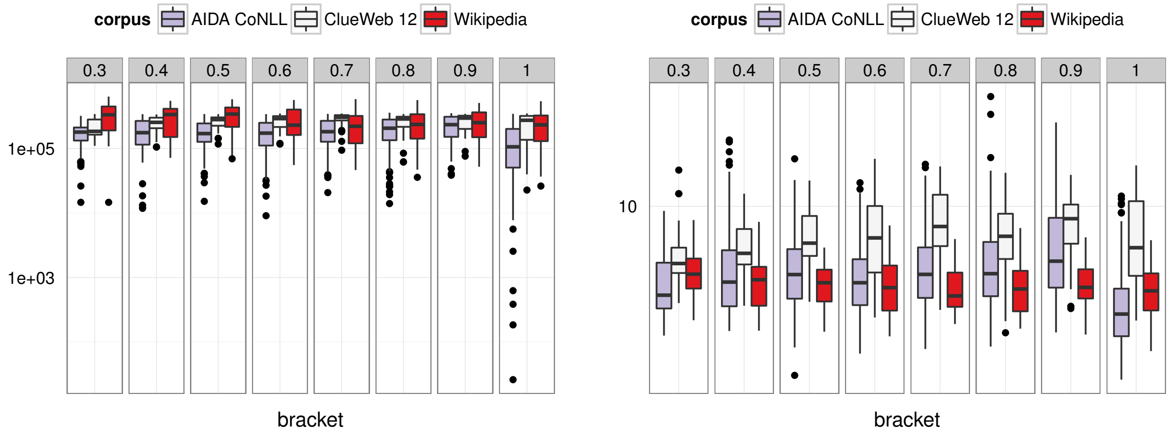
Fig. 2. Corpus statistics.

ument level) may not be good predictors of actual performance in real applications. For instance, if a large fraction of the mentions in the benchmarks are "too easy" compared to real documents, the metrics will overestimate the true accuracy.

Of course, in order to fine tune the difficulty of the mentions and the documents in a benchmark one needs a reliable indicator of "difficulty" that can be applied to a large number of documents. Manual annotations are clearly undesirable here, and so is *crowdsourcing*: the number of annotations needed might prove prohibitive and even if resources are not a concern this leads to a *single* benchmark (i.e., if more documents are needed, more annotations would be required).

4.3. New Benchmarks

To obtain new and balanced benchmarks, we consider the PRIOR baseline as a proxy for the true difficulty of a mention, and we obtain documents by sampling from large publicly annotated corpora such as ClueWeb and Wikipedia. In this way, we can easily collect large corpora of previously annotated documents and retain as many as needed while tuning disambiguation difficulty to the desired proportion.



(a) Number of nodes.

(b) Average node degree.

Fig. 3. Disambiguation graph statistics.

More precisely, we applied PRIOR to all documents of Wikipedia (20130606 dump) and the FACC1 annotated ClueWeb 2012 dataset [2]. We grouped documents by the resulting average accuracy (of all mentions in the document), and randomly picking 40 documents for each bracket. Also, we further restricted the benchmarks to documents in which PRIOR achieved 0.3 or higher accuracy as we observed that below that threshold, the quality of the annotations in the ClueWeb dataset were very low. Finally, we controlled the number of mentions per document: for the Wikipedia corpus we have the mean at 20.8 ($\sigma = 4.9$) and for the ClueWeb 2012 we have the mean at 35.5 ($\sigma = 8.5$).

Some statistics about the proposed benchmarks: Fig. 2a shows the average number of mentions per document and Fig. 2b shows the average number of candidates per mention. For the sake of comparison, we also report the same statistics from the documents in the AIDA-CONLL dataset in the respective accuracy brackets. Fig. 3 shows statistics about the disambiguation graphs built by our method (which, as discussed in Section 3, depend both on the number of candidates per mention and on how densely connected they are in the entity graph). Fig. 3a shows the average graph sizes (in terms of number of nodes) and Fig. 3b shows the average node degree.

As one can see, the variability in our datasets is considerably smaller compared to AIDA-CONLL, particularly when it comes to clear outliers (indicated as individual dots in the charts).

Table 3

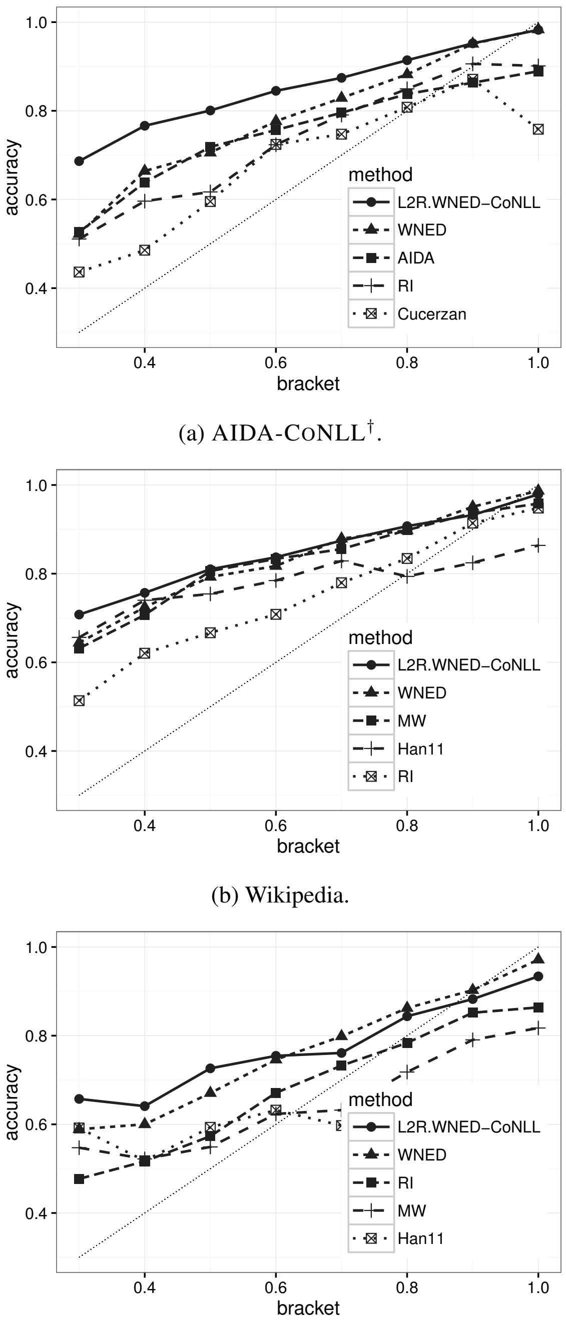
Average per-bracket accuracy on large-scale benchmarks. Brackets for AIDA-CONLL are as in Table 2; only those brackets with PRIOR accuracy 0.3 or higher were used.

Method	AIDA-CONLL	Wikipedia	ClueWeb 12
PRIOR	0.57	0.56	0.57
CONTEXT	0.39	0.59	0.42
Cucerzan	0.68	0.66	0.60
M&W	0.58	0.83	0.65
Han11	0.57	0.78	0.61
AIDA	0.75	0.63	0.59
GLOW	0.61	0.69	0.57
RI	0.74	0.75	0.68
WNED	0.79	0.84	0.77
L2R.WNED	**0.85**	**0.85**	**0.78**

4.4. *Results on the New Benchmarks*

Fig. 4 shows the accuracy on the new benchmarks. We plot the accuracy of the best performing methods for each of the difficulty brackets (defined by the accuracy of the PRIOR baseline). For clarity, we plot the accuracy of the best 5 approaches. For comparison, we also show the accuracy of each method on the AIDA-CONLL benchmark. For the Wikipedia and ClueWeb benchmarks, each bracket corresponds to exactly 40 documents, whereas for the AIDA-CONLL dataset

[2] http://lemurproject.org/clueweb12/FACC1/



(a) AIDA-CoNLL†.

(b) Wikipedia.

(c) ClueWeb 12.

Fig. 4. Average accuracy of the top-5 methods on the Wikipedia, Clueweb 12, and AIDA-CoNLL datasets grouped by the accuracy of the PriorProb baseline.

the brackets are as in Table 2. For convenience, a diagonal dotted line whose area under the curve (AUC) is 0.5 (loosely corresponding to the PRIOR baseline) is also shown. Methods consistently above that line are expected to outperform the PRIOR baseline in practice. Table 3 shows the average accuracy of every method across brackets, corresponding to the AUC in Fig. 4.

A few observations are worth mentioning here. First, the two new benchmarks complement the AIDA-CoNLL benchmark: overall, the Wikipedia benchmark is easier than AIDA-CoNLL, while the ClueWeb 12 is harder. Second, as before, the RI method performed very well, although not as dominantly as in the four public benchmarks. It also seems that the previous supervised methods tend to over-perform on their own development datasets (Wikipedia for M&W and CoNLL for AIDA).

Our L2R.WNED and WNED systems outperform all other competitors across all benchmarks, performing much better on the more “difficult” cases (i.e., in lower brackets). In concrete terms, WNED and L2R.WNED exhibit, on average, 21% and 26% relative gain in accuracy over the previous methods (excluding the baselines) on the three benchmarks combined, which is significant. Given that our development and tuning was done with a subset of the AQUAINT, MSNBC and ACE2004, the strong results of WNED and L2R.WNED demonstrate the robustness and generality of our approach.

4.5. Qualitative Error Analysis

We now look at the kinds of errors made by our method. To do so, we manually inspected every error for the smaller MSNBC, AQUAINT, and ACE2004 datasets, and analyzed 20 errors randomly picked in each bracket for the larger ones.

The first observation is that in the older benchmarks, a larger fraction of the errors in our method happen in the candidate selection phase, as illustrated in Fig. 5. On average, 54% of the errors in the smaller benchmarks are due to candidate selection (compared to 18% in the other ones). This reinforces the hypothesis that the entities mentioned in these older benchmarks are easier to disambiguate[3].

[3] Note: given that the smaller benchmarks are older, it is unlikely they mention more *out-of-KB* entities than the other ones, especially AIDA-CoNLL. Thus, because we use the same standard NLP pipeline for processing the inputs across all benchmarks, the discrepancy in Fig. 5 can only mean that our method is successful on

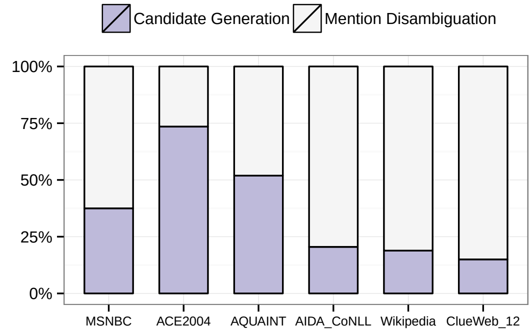


Fig. 5. Breakdown of errors by WNED across benchmarks; for AIDA-CoNLL, Wikipedia and ClueWeb 12, the errors are estimated from a sample.

Below we discuss prototypical errors in each of the phases.

Errors during Candidate Selection

Incorrect Co-reference Resolution We employ a co-reference resolution algorithm in our text processing pipeline to increase recall. Due to the heuristic nature of the algorithm, it is possible that distinct named entities are incorrectly deemed to be the same. For example, in the sentence

> "Time Warner stagnated for five years after it was created in 1990 by the merger of Time and Warner."

the entity "Time" at the end of the sentence is incorrectly resolved to "Time Warner", leading to an error. About 1% of the errors (1.5% in the harder benchmarks) are due to incorrect resolution of named entities.

Incomplete Alias Dictionary Currently, we disambiguate only those mentions corresponding to an alias from Wikipedia, leading to problems in sentences like

> "Thirteen miners were trapped inside the Sago Mine near Buckhannon, W. Va."

In this case we miss the abbreviation "W. Va." for West Virginia. This kind of error was noticeably more common in the easier benchmarks (accounting for 30% of the errors in the ACE2004 dataset). In the AIDA-CoNLL benchmark only 2% of the errors are due to this problem.

Aggressive Pruning Another source of error by our method is pruning the correct entity from the disambiguation graph. For example, in sentence

> "A state coordinator for the Florida Green Party said she had been ..."

the correct entity (the *Green Party of Florida*) is pruned due its low prior but could probably be correctly resolved given the mention to *Florida* in the same sentence. Instead, WNED links the mention to the US Green Party. Of course, pruning is done to reduce the cost of the random walks, and future algorithmic improvements can alter this trade-off.

Errors during Mention Disambiguation

These are errors where the correct entities according to the ground truth were selected as the candidates but not chosen during the mention disambiguation phase by our algorithm.

Lack of Application Domain We observed that most of the errors associated with locations happen because the documents in most benchmarks are news articles that start with the location of the news source reporting the news (e.g., New York Times documents always start with a mention to New York). More often than not, such locations are totally unrelated to the topic of the documents and other mentions in the document, breaking the global coherence assumption. These errors, which can be easily fixed via pre-processing, accounts for 5% of the mistakes of our algorithm in the MSNBC and AIDA benchmarks and 2% across all benchmarks.

Need for Deeper Text Analysis There are of course very hard disambiguation cases where a deeper understanding of the text would be needed for a successful algorithmic approach. One example is the sentence:

> "Maj. Gen. William Caldwell, a U.S. military spokesman, told reporters that ..."

In this case there are two candidates with the same name and high military rank, thus being semantically related to the document and confusing the algorithm. In this case, extraneous facts about the candidates, unrelated to the text itself, could be used for disambiguating the mention. For instance the candidate incorrectly chosen by our algorithm died in the 1820s while the correct candidate was still alive at the time the benchmark article was written. Given that the doc-

most of the (*in-KB*) entities in the older benchmarks, making them "easier" than the other ones.

Table 4
Questionable disambiguation errors

Mention	WNED Suggestion	Ground Truth
Iraqi	Iraqi people	Iraq
Hungarian	Hungary	Hungarian people
executives	Corporate title	Chief executive officer
Russian	Russian language	Russians
Iranian	Iranian people	Iran
Greek	Greek language	Ancient Greece
civil war	American Civil War	Civil war
broadcaster	Presenter	Broadcasting

ument states the facts as current news, the incorrect candidate could have been pruned out.

Questionable Errors

We argue that in many cases, our algorithm (as well as other systems) chose an entity that are considered erroneous by the ground truth but that would be acceptable to a human judge. For example, in the sentence:

> "Coach Saban said the things Crimson Tide fans most wanted to hear."

our system links "Crimson Tide" in the sentence to the *Alabama Crimson Tide football*, which is the *men's* varsity football team of the university while the ground truth refers to *Alabama Crimson Tide* which corresponds to both the men's and women's teams. We found that about 17% of the errors are in this category, with a higher prevalence in the harder benchmarks (21%). Table 4 lists many other similar errors, where a case can be made that the ground-truth itself is probably too strict.

Impact of the Greedy Approach

Given the iterative WNED is a greedy algorithm, it is interesting to see how an erroneous disambiguation decision influences future ones, especially in the very first round. In all benchmarks, we found one error in the first round among all the errors in MSNBC, AQUAINT and ACE2004 datasets[4], and less than eight errors from the random samples in the other 3 benchmarks. In all cases, the first error did not prevent the algorithm from correctly disambiguating other mentions.

As for the initialization step, we found that most documents in our benchmarks do have unambiguous mentions available, and most of them are correctly linked to the true entity[5]. In MSNBC, we have 2 errors from the unambiguous mentions, *New york stock exchange* and *NYSE*, both are linked to *New york mercantile exchange*. This error barely affects the semantic signature since they are still stock related entities. There are 5 such errors in AQUAINT, and 1 error in ACE2004, all of which have little affect on the linking results of other mentions in the same document.

Finally, we found that most other errors happened after 5 iterations, when the document disambiguation vector already captures the topic fairly well. These errors are for mentions that are not semantically related to other mentions in the document, or simply due to the disproportionately high priors favoring (incorrectly) head entities.

5. Related Work

NED is commonly cast as a ranking problem where we estimate the likelihood with which each candidate entity should be assigned to each mention, choosing the one with the highest likelihood. Based on the features used and the way mentions are disambiguated, most work about entity disambiguation in the literature can be divided into two main groups.

The first group, which we refer to as *local methods*, disambiguate each mention in a document independently of others using local features. These methods represent mentions and entities with feature vectors and measure the likelihood using a compatibility function such as vector similarity measures. The most commonly used local features include lexical features such as bag-of-words [2], keyphrases [15], n-grams [37], or semantic word embeddings [47] extracted or derived from the surrounding context, and statistical features such as the probability of entities given a mention from a knowledge graph. While the first local methods were unsupervised [2,3], combining features through manually tuned parameters, others optimize the parameter selection with various classifiers [8,24,26,43,44,46] often using Wikipedia as a source of training data. As described, the main issues with local methods are the data sparsity problem on the features and ignoring the semantic dependencies between mentions.

The second group of methods, which we refer to as *global methods*, adhere to the hypothesis that men-

[4] A mention to the *USS Cole* which should have been linked to *USS Cole (DDG-67)*, was linked to *USS Cole bombing*.

[5] Recall (Sec. 3) we initialize the document disambiguation vector with unambiguous mentions when available.

tions from the same document are semantically coherent around the topic of the document, and cast the disambiguation problem as an optimization whose objective is to find the assignment with maximum global coherence. Thus appropriate semantic relatedness measures and efficient inference for the assignment with maximum global coherence are the two most important factors. Cucerzan [7] proposed the first global approach which measures the semantic relatedness between entities using Wikipedia categories of entities. Milne and Witten (M&W) [26] represent each entity with their directly connected entities in a knowledge graph and measure relatedness using normalized Google distance between the representations. Kulkarni et al. [20] also use the M&W semantic relatedness measure in their collective approach. In addition to that, Ratinov et al. [31] add the PMI (Pointwise Mutual Information) of entities into their SVM classifier, and Cheng and Roth [6] model more fine-grained semantics such as relations between mentions as constraints in the semantic relatedness, which greatly improved the results. More recently, Zwicklbauer et al. [47] use word embeddings from the context of entities as the representation to measure their semantic relatedness.

With semantic relatedness measures, entities and their relationships can be represented as an entity graph, and the optimization objective is to find a subgraph with maximum global coherence in which each node corresponds to the referent entity of a mention in the document. AIDA [14] aims to find a dense subgraph in the mention-entity graph that contains all mentions and a single mention-entity edge for each mention according to their definition of graph density. Han et. al [13] also use the graph built around candidates of mentions and apply random walk over the graph to obtain the pagerank score of entities which is used to measure the global coherence. Instead of using PageRank, AGDISTIS [39] applies the HITS algorithm on the graph and uses the authority score of a candidate for its global coherence with mentions and other entities.

The assumption that mentions belong to one single topic may not be true in many documents. For such cases, topic modelling can be used to distribute latent topics across candidate entities. Han and Sun [12] combine the local compatibility and global coherence using a generative entity-topic model to infer the underlying referent entities. Li et al. [22] introduce additional information of entities mined from external corpus into a generative graph model to improve the effectiveness of NED, especially for entities with rare information available in the knowledge graph. Kataria et al. [18] proposed a semi-supervised hierarchical topic model for entity disambiguation based on the Wikipedia's category hierarchy and word-entity associations. Ganea et al. [9] recently proposed a lightweight probabilistic model based on the co-occurrence statistics derived from knowledge graphs.

Most semantic relatedness measures employed in NED systems compute the relatedness between two entities only. Instead, we propose a unified semantic representation for both entities and the collective entity-to-mention assignment (Γ), with which we can measure how well a candidate entity fits with those previously assigned in an unified way. Our representation can capture the semantics of unpopular entities (those with low degree in the entity graph), which makes our NED approach more robust. This observation is supported by our experiments. The idea of using random walks with restart has been applied on graphs constructed from the WordNet [25], with the stationary distribution to represent the semantics of words. It has been shown to be effective in the word similarity measurement [1,16], and word sense disambiguation [30]. However, we are not aware of any previous work using the stationary distribution from random walk with restart to represent entities and documents in NED.

When it comes to learning to rank, Zheng et.al [45] applied learning to rank approaches on the NED task and demonstrated its superior effectiveness over most state-of-the-art algorithms. Their results showed that the *listwise* method ListNet [5] performed better than the *pairwise* approach Ranking Perceptron [34]. However, we found that the pairwise approach LambdaMART [41] achieved the best performance on our datasets among most learning to rank algorithms.

6. Conclusion

We described a method for named entity disambiguation that combines lexical and statistical features with *semantic* signatures derived from random walks over suitably designed disambiguation graphs. Our semantic representation uses more relevant entities from the knowledge graph, thus reducing the effect of feature sparsity, and results in substantial accuracy gains. We described a hand-tuned greedy algorithm as well as one based on learning-to-rank. Both outperform the previous state-of-the-art by a wide margin. Moreover, we showed that our L2R.WNED algorithm trained on the standard AIDA-CoNLL corpus is quite robust across benchmarks. We also evaluated both systems using the GERBIL framework on 16 public datasets and showed the superiority of our approach.

Moreover, we demonstrated several shortcomings of the existing NED benchmarks and described an effective way for deriving *better* benchmarks and described two new such benchmarks based on web-scale annotated corpora (ClueWeb12 and Wikipedia). Our benchmark generation method can be tuned to produce "harder" or "easier" cases as desired. Overall, the benchmarks we describe complement the largest currently available public benchmark. Our experimental evaluation compared our methods against six leading competitors and two very strong baselines, revealing the superiority and robustness of our NED system in a variety of settings. Our method was particularly robust when disambiguating unpopular entities, making it a good candidate to address the "long tail" in Information Extraction.

Future work There are several directions worth exploring. Sec. 4.5 lists several ideas for algorithmic improvements that can lead to better NED systems in the future. Also, while the new benchmarks described here can be used for both accuracy and scalability tests (as one can easily obtain large quantities of documents from ClueWeb12 and Wikipedia), further work is needed in helping the design and verification of ground-truths.

System performance is one issue we will need to improve. Currently, the average time to disambiguate a document with less than 100 is in the order of a few minutes, and the time could increase with the number of mentions in a document and average number of candidates per mention. Majority of the time is spent on the expensive random walk computations. Approximating the semantic signature with less expensive random walk algorithms would be helpful. Other than that, designing a system using appropriate indexing and utilizing the current large-scale data processing infrastructure would also be interesting.

References

[1] Eneko Agirre, Enrique Alfonseca, Keith B. Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*, pages 19–27. ACL, 2009. URL http://www.aclweb.org/anthology/N09-1003.

[2] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In Christian Boitet and Pete Whitelock, editors, *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.*, pages 79–85. Morgan Kaufmann Publishers / ACL, 1998. URL http://aclweb.org/anthology/P/P98/P98-1012.pdf.

[3] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In Diana McCarthy and Shuly Wintner, editors, *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. ACL, 2006. URL http://aclweb.org/anthology/E/E06/E06-1002.pdf.

[4] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 193–200. MIT Press, 2006. URL http://papers.nips.cc/paper/2971-learning-to-rank-with-nonsmooth-cost-functions.

[5] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In Zoubin Ghahramani, editor, *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 129–136. ACM, 2007. DOI https://doi.org/10.1145/1273496.1273513.

[6] Xiao Cheng and Dan Roth. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1787–1796. ACL, 2013. URL http://aclweb.org/anthology/D/D13/D13-1184.pdf.

[7] Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In Jason Eisner, editor, *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 708–716. ACL, 2007. URL http://www.aclweb.org/anthology/D07-1074.

[8] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 277–285. Tsinghua University Press, 2010. URL http://aclweb.org/anthology/C10-1032.

[9] Octavian-Eugen Ganea, Marina Ganea, Aurélien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao, editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 927–938. ACM, 2016. DOI https:

//doi.org/10.1145/2872427.2882988.

[10] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. ISBN 0-7167-1044-7.

[11] Zhaochen Guo and Denilson Barbosa. Robust entity linking via random walks. In Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang, editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 499–508. ACM, 2014. DOI https://doi.org/10.1145/2661829.2661887.

[12] Xianpei Han and Le Sun. An entity-topic model for entity linking. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 105–115. ACL, 2012. URL http://www.aclweb.org/anthology/D12-1010.

[13] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In Wei-Ying Ma, Jian-Yun Nie, Ricardo A. Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft, editors, *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 765–774. ACM, 2011. DOI https://doi.org/10.1145/2009916.2010019.

[14] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 782–792. ACL, 2011. URL http://www.aclweb.org/anthology/D11-1072.

[15] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. KORE: keyphrase overlap relatedness for entity disambiguation. In Xue-wen Chen, Guy Lebanon, Haixun Wang, and Mohammed J. Zaki, editors, *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 545–554. ACM, 2012. DOI https://doi.org/10.1145/2396761.2396832.

[16] Thad Hughes and Daniel Ramage. Lexical semantic relatedness with random graph walks. In Jason Eisner, editor, *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 581–589. ACL, 2007. URL http://www.aclweb.org/anthology/D07-1061.

[17] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1148–1158. ACL, 2011. URL http://www.aclweb.org/anthology/P11-1115.

[18] Saurabh Kataria, Krishnan S. Kumar, Rajeev Rastogi, Prithviraj Sen, and Srinivasan H. Sengamedu. Entity disambiguation with hierarchical topic models. In Chid Apté, Joydeep Ghosh, and Padhraic Smyth, editors, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1037–1045. ACM, 2011. DOI https://doi.org/10.1145/2020408.2020574.

[19] Grzegorz Kondrak. *N*-gram similarity and distance. In Mariano P. Consens and Gonzalo Navarro, editors, *String Processing and Information Retrieval, 12th International Conference, SPIRE 2005, Buenos Aires, Argentina, November 2-4, 2005, Proceedings*, volume 3772 of *Lecture Notes in Computer Science*, pages 115–126. Springer, 2005. DOI https://doi.org/10.1007/11575832_13.

[20] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 457–466. ACM, 2009. DOI https://doi.org/10.1145/1557019.1557073.

[21] Hang Li. *Learning to Rank for Information Retrieval and Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011. DOI https://doi.org/10.2200/S00348ED1V01Y201104HLT012.

[22] Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. Mining evidences for named entity disambiguation. In Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy, editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 1070–1078. ACM, 2013. DOI https://doi.org/10.1145/2487575.2487681.

[23] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In Chiara Ghidini, Axel-Cyrille Ngonga Ngomo, Stefanie N. Lindstaedt, and Tassilo Pellegrini, editors, *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, ACM International Conference Proceeding Series, pages 1–8. ACM, 2011. DOI https://doi.org/10.1145/2063518.2063519.

[24] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In Mário J. Silva, Alberto H. F. Laender, Ricardo A. Baeza-Yates, Deborah L. McGuinness, Bjørn Olstad, Øystein Haug Olsen, and André O. Falcão, editors, *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 233–242. ACM, 2007. DOI https://doi.org/10.1145/1321440.1321475.

[25] George A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. DOI https://doi.org/10.1145/219717.219748.

[26] David N. Milne and Ian H. Witten. Learning to link with wikipedia. In James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury, editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 509–518. ACM, 2008. DOI https://doi.org/10.1145/1458082.1458150.

[27] David N. Milne and Ian H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Razvan Bunescu, Evgeniy Gabrilovich, and Rada Mihalcea, editors, *Wikipedia and Artificial Intelligence: An Evolving Synergy, Papers from the 2008 AAAI Workshop, Chicago, Illinois, USA, July 13âĂŞ14, 2008*, volume WS-08-15 of *AAAI Workshops*. AAAI Press, 2008. URL http://www.aaai.org/Papers/Workshops/2008/WS-08-15/WS08-15-005.pdf.

[28] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014. URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/291.

[29] Francesco Piccinno and Paolo Ferragina. From tagme to WAT: a new entity annotator. In David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang, editors, *ERD'14, Proceedings of the First ACM International Workshop on Entity Recognition & Disambiguation, July 11, 2014, Gold Coast, Queensland, Australia*, pages 55–62. ACM, 2014. DOI https://doi.org/10.1145/2633211.2634350.

[30] Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1341–1351. ACL, 2013. URL http://aclweb.org/anthology/P/P13/P13-1132.pdf.

[31] Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1375–1384. ACL, 2011. URL http://www.aclweb.org/anthology/P11-1138.

[32] Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy. Benchmarking the extraction and disambiguation of named entities on the Semantic Web. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 4593–4600. European Language Resources Association (ELRA), 2014. URL http://www.lrec-conf.org/proceedings/lrec2014/summaries/176.html.

[33] Felix Sasaki, Tatiana Gornostay, Milan Dojchinovski, Michele Osella, Erik Mannens, Giannis Stoitsis, Phil Ritchie, Thierry Declerck, and Kevin Koidl. Introducing FREME: deploying linguistic linked data. In Jorge Gracia, John P. McCrae, and Gabriela Vulcu, editors, *Proceedings of the Fourth Workshop on the Multilingual Semantic Web (MSW4) co-located with 12th Extended Semantic Web Conference (ESWC 2015), Portorož, Slovenia, June 1, 2015.*, volume 1532 of *CEUR Workshop Proceedings*, pages 59–66. CEUR-WS.org, 2015. URL http://ceur-ws.org/Vol-1532/paper6.pdf.

[34] Libin Shen and Aravind K. Joshi. Ranking and reranking with perceptron. *Machine Learning*, 60(1-3):73–96, 2005. DOI https://doi.org/10.1007/s10994-005-0918-9.

[35] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UMASS-CS-2012-015, Department of Computer Science, University of Massachusetts, Amherst, 2012. URL https://web.cs.umass.edu/publication/docs/2012/UM-CS-2012-015.pdf.

[36] René Speck and Axel-Cyrille Ngonga Ngomo. Named entity recognition using FOX. In Matthew Horridge, Marco Rospocher, and Jacco van Ossenbruggen, editors, *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, volume 1272 of *CEUR Workshop Proceedings*, pages 85–88. CEUR-WS.org, 2014. URL http://ceur-ws.org/Vol-1272/paper_70.pdf.

[37] Nadine Steinmetz and Harald Sack. Semantic multimedia information retrieval based on contextual descriptions. In Philipp Cimiano, Óscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science*, pages 382–396. Springer, 2013. DOI https://doi.org/10.1007/978-3-642-38288-8_26.

[38] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pages 613–622. IEEE Computer Society, 2006. DOI https://doi.org/10.1109/ICDM.2006.70.

[39] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. AGDISTIS - agnostic disambiguation of named entities using linked open data. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1113–1114. IOS Press, 2014. DOI https://doi.org/10.3233/978-1-61499-419-0-1113.

[40] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. GERBIL: general entity annotator benchmarking framework. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1133–1143. ACM, 2015. DOI https://doi.org/10.1145/2736277.2741626.

[41] Qiang Wu, Christopher J. C. Burges, Krysta Marie Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010. DOI https://doi.org/10.1007/s10791-009-9112-1.

[42] Lei Zhang and Achim Rettinger. X-LiSA: Cross-lingual semantic annotation. *Proceedings of the VLDB Endowment*, 7 (13):1693–1696, 2014. URL http://www.vldb.org/pvldb/vol7/p1693-zhang.pdf.

[43] Wei Zhang, Jian Su, Chew Lim Tan, and Wenting Wang. Entity linking leveraging automatically generated annotation. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING 2010,*

23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China, pages 1290–1298. Tsinghua University Press, 2010. URL http://aclweb.org/anthology/C10-1145.

[44] Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan. Entity linking with effective acronym expansion, instance selection, and topic modeling. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1909–1914. IJCAI/AAAI, 2011. DOI https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-319.

[45] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 483–491. ACL, 2010. URL http://www.aclweb.org/anthology/N10-1072.

[46] Yiping Zhou, Lan Nie, Omid Rouhani-Kalleh, Flavian Vasile, and Scott Gaffney. Resolving surface forms to Wikipedia topics. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 1335–1343. Tsinghua University Press, 2010. URL http://aclweb.org/anthology/C10-1150.

[47] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Robust and collective entity disambiguation through semantic embeddings. In Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel, editors, *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 425–434. ACM, 2016. DOI https://doi.org/10.1145/2911451.2911535.

Appendix

A. Evaluation on GERBIL

GERBIL [40] is a general framework for entity annotation, which has more than 11 NED systems and 16 public datasets for the entity disambiguation task. We compare our system with all available systems including graph-based systems: AGDISTIS [39], AIDA [14], Babelfy [28], FOX [36], WAT [29], xLisa [42], and PBoH [9]; context-based systems: DBpedia Spotlight [23], FREME NER [33], Kea [37], and NERD-ML [32]. The datasets mainly contain documents from news article, RSS feeds, tweets, encyclopedia and the mix. Detailed statistics of the datasets are shown in Table 5. We can see that most datasets are from news articles which means that entities mentioned in the documents are popular ones. Performance on datasets with very few mentions per document such as *Microposts2014* and *N3-RSS-500* will depends more on the contextual similarity, and less on the global coherence of most approaches.

Table 5
Statistics of datasets in GERBIL [40].

corpus	topic	#docs	#mentions/doc
ACE2004	news	57	4.44
AQUAINT	news	50	14.54
MSNBC	news	20	32.50
AIDA/CoNLL	news	1393	19.97
DBpediaSpotlight	news	58	5.69
KORE50	mixed	50	2.86
Microposts2014	tweets	3505	0.65
N3-RSS-500	RSS-feeds	500	0.99
N3-Reuters-128	news	128	4.85
OKE 2015 Task 1	encyclopedia	199	5.41

Table 6 gives the results evaluated using GERBIL [6], PBoH [9] [7], and our systems [8]. Note that the results of PBoH in Table 6 are from their updated report on GERBIL 1.2.4, which is different from results reported in [9] [9].

We can see that comparing to other NED systems, our two systems WNED and L2R.WNED-CoNLL (trained with 20% AIDA/CoNLL dataset) can achieve very competitive results on most of the datasets. Although no special processing is applied on the micropost2014 datasets, our systems still achieve better results than all other systems except the PBoH. One main types of errors on microposts are from the candidate selection due to the casual writing style in micropost which causes many uncommon name variations.

[6] http://gerbil.aksw.org/gerbil/experiment?id=201611040001

[7] http://gerbil.aksw.org/gerbil/experiment?id=201610270004

[8] Due to an implementation issue, each dataset has to be evaluated separately. Results are available in http://dx.doi.org/10.7939/DVN/10968

[9] There is a drop on the results from version 1.1.4 to version 1.2.4. Details: https://github.com/AKSW/gerbil/issues/98

Datasets	AGDISTIS [39]	AIDA [14]	Babelfy [28]	DBpedia Spotlight [23]	FOX [36]	FREME NER [33]	Kea [37]	NERD-ML [32]	WAT [29]	xLisa [42]	PBoH [9]	WNED	L2R.WNED-CoNLL
ACE2004	0.65	0.69	0.53	0.48	0.00	0.49	0.66	0.58	0.66	0.70	0.72	**0.77**	**0.76**
	0.77	0.82	0.70	0.68	0.37	0.65	0.77	0.73	0.77	0.80	0.83	**0.88**	**0.87**
	0.66	0.80	0.61	0.58	0.00	0.58	0.76	0.67	0.76	**0.81**	0.79	**0.83**	**0.81**
	0.78	0.89	0.76	0.75	0.39	0.71	0.84	0.79	0.85	0.88	0.86	**0.91**	**0.90**
AQUAINT	0.52	0.55	0.68	0.53	0.00	0.56	0.78	0.60	0.73	0.76	**0.81**	**0.79**	**0.79**
	0.51	0.55	0.68	0.52	0.00	0.43	0.78	0.58	0.74	0.75	**0.81**	**0.79**	**0.79**
	0.73	0.57	0.70	0.55	0.00	0.58	0.81	0.62	0.75	0.79	**0.84**	**0.83**	**0.83**
	0.59	0.56	0.70	0.54	0.00	0.44	**0.80**	0.60	0.76	0.77	**0.83**	**0.83**	**0.83**
MSNBC	0.73	0.69	0.71	0.42	0.02	0.22	0.78	0.62	0.73	0.50	**0.82**	**0.88**	**0.88**
	0.73	0.65	0.68	0.44	0.02	0.16	0.77	0.64	0.73	0.50	0.82	**0.90**	**0.89**
	0.74	0.74	0.76	0.46	0.02	0.24	0.84	0.67	0.79	0.55	**0.86**	**0.89**	**0.89**
	0.73	0.70	0.73	0.48	0.02	0.18	0.84	0.70	0.80	0.57	0.85	**0.91**	**0.90**
AIDA/CoNLL-Complete	0.55	0.68	0.66	0.50	0.51	0.38	0.61	0.20	0.71	0.47	0.75	**0.76**	**0.77**
	0.53	0.66	0.60	0.50	0.48	0.29	0.57	0.12	0.68	0.45	0.75	**0.76**	**0.77**
	0.57	0.77	0.74	0.58	0.54	0.44	0.68	0.24	**0.80**	0.54	**0.80**	**0.79**	**0.80**
	0.52	0.76	0.68	0.59	0.50	0.33	0.65	0.14	**0.78**	0.52	**0.78**	**0.78**	**0.79**
AIDA/CoNLL-Test A	0.54	0.67	0.65	0.48	0.49	0.28	0.61	0.00	0.70	0.45	**0.75**	**0.76**	**0.76**
	0.50	0.62	0.59	0.47	0.45	0.23	0.56	0.00	0.66	0.41	**0.73**	**0.75**	**0.75**
	0.56	0.74	0.74	0.55	0.53	0.33	0.67	0.00	0.78	0.52	**0.80**	0.78	**0.79**
	0.49	0.71	0.68	0.55	0.47	0.25	0.64	0.00	**0.76**	0.48	**0.77**	0.75	**0.76**
AIDA/CoNLL-Test B	0.54	0.69	0.68	0.52	0.49	0.35	0.61	0.01	0.72	0.47	**0.75**	**0.75**	**0.76**
	0.54	0.68	0.62	0.51	0.48	0.22	0.61	0.00	0.70	0.46	0.75	**0.76**	**0.77**
	0.55	0.77	0.76	0.60	0.52	0.40	0.69	0.00	**0.80**	0.54	**0.80**	0.77	**0.79**
	0.54	0.78	0.70	0.60	0.51	0.26	0.70	0.01	**0.80**	0.53	**0.79**	0.78	**0.79**
AIDA/CoNLL-Training	0.55	0.69	0.65	0.50	0.52	0.39	0.61	0.28	0.71	0.48	0.75	**0.76**	**0.77**
	0.53	0.66	0.60	0.50	0.50	0.30	0.56	0.17	0.68	0.45	**0.73**	**0.77**	**0.77**
	0.57	0.77	0.74	0.58	0.55	0.45	0.69	0.33	**0.81**	0.56	**0.80**	0.79	**0.80**
	0.52	0.76	0.68	0.59	0.51	0.35	0.64	0.21	**0.79**	0.53	**0.78**	**0.78**	**0.79**
DBpediaSpotlight	0.27	0.25	0.52	0.71	0.15	0.45	0.74	0.56	0.67	0.71	**0.79**	**0.79**	**0.80**
	0.28	0.21	0.51	0.69	0.12	0.31	0.73	0.53	0.69	0.71	0.80	**0.81**	**0.82**
	0.40	0.25	0.52	0.71	0.15	0.45	0.74	0.56	0.67	0.71	**0.80**	**0.80**	**0.81**
	0.36	0.21	0.51	0.69	0.12	0.31	0.73	0.53	0.69	0.71	0.80	**0.82**	**0.83**
KORE50	0.33	**0.69**	**0.74**	0.46	0.27	0.17	0.60	0.31	0.62	0.51	0.63	0.56	0.50
	0.30	**0.64**	**0.70**	0.42	0.22	0.14	0.53	0.25	0.52	0.45	0.58	0.52	0.50
	0.33	**0.69**	**0.74**	0.46	0.27	0.17	0.60	0.31	0.62	0.51	0.63	0.56	0.50
	0.30	**0.64**	**0.70**	0.42	0.22	0.14	0.53	0.25	0.52	0.45	0.59	0.52	0.50
Microposts2014-Test	0.33	0.42	0.48	0.50	0.22	0.42	0.64	0.52	0.60	0.55	**0.73**	0.63	**0.67**
	0.60	0.59	0.63	0.66	0.49	0.60	0.76	0.67	0.74	0.68	**0.85**	0.75	**0.79**
	0.42	0.42	0.48	0.50	0.22	0.42	0.64	0.52	0.60	0.55	**0.74**	0.65	**0.69**
	0.61	0.59	0.63	0.66	0.49	0.60	0.76	0.67	0.74	0.68	**0.85**	0.76	**0.79**
Microposts2014-Train	0.42	0.51	0.51	0.48	0.31	0.46	0.65	0.52	0.63	0.59	**0.71**	0.64	**0.67**
	0.61	0.61	0.61	0.61	0.48	0.56	0.74	0.63	0.73	0.67	**0.81**	0.74	**0.76**
	0.51	0.51	0.51	0.48	0.31	0.46	0.65	0.52	0.63	0.59	**0.73**	0.67	**0.70**
	0.63	0.61	0.61	0.61	0.48	0.56	0.74	0.63	0.73	0.67	**0.82**	0.75	**0.78**
N3-RSS-500	0.61	0.45	0.44	0.20	0.56	0.28	0.44	0.38	0.44	0.45	0.53	**0.69**	**0.68**
	0.61	0.39	0.38	0.16	0.54	0.20	0.39	0.30	0.37	0.38	0.53	**0.69**	**0.68**
	0.52	**0.66**	0.64	0.32	0.50	0.44	0.62	0.57	0.64	**0.65**	0.55	**0.65**	0.63
	0.52	**0.64**	0.63	0.41	0.49	0.45	0.61	0.58	0.63	**0.66**	0.48	0.62	0.61
N3-Reuters-128	**0.66**	0.47	0.45	0.33	0.54	0.24	0.51	0.41	0.52	0.39	**0.65**	0.63	0.64
	0.72	0.38	0.39	0.27	0.57	0.16	0.46	0.35	0.44	0.34	**0.72**	**0.63**	**0.63**
	0.64	0.57	0.55	0.41	0.52	0.31	0.61	0.51	0.63	0.49	**0.69**	0.62	**0.65**
	0.68	0.51	0.55	0.41	0.54	0.29	0.60	0.51	0.59	0.52	**0.72**	0.60	0.60
OKE 2015 Task 1 evaluation dataset	0.59	0.56	0.59	0.31	0.56	0.32	**0.63**	0.61	0.57	**0.62**	**0.63**	**0.62**	**0.62**
	0.60	0.55	0.58	0.27	0.53	0.26	**0.63**	0.60	0.56	0.61	**0.63**	**0.62**	**0.62**
	0.62	0.63	0.66	0.36	0.60	0.38	**0.71**	**0.70**	0.65	**0.71**	0.68	0.65	0.65
	0.61	0.62	0.65	0.30	0.56	0.28	**0.71**	0.68	0.62	**0.70**	0.67	0.64	0.65
OKE 2015 Task 1 example set	**1.00**	0.60	0.4	0.22	**0.78**	0.25	0.55	0.00	0.60	0.5	0.50	0.67	0.67
	1.00	0.72	0.65	0.44	0.67	0.44	0.69	0.33	0.72	0.69	0.67	**0.75**	**0.75**
	1.00	**0.86**	0.57	0.50	0.80	0.40	0.75	0.00	**0.86**	0.80	0.67	0.75	0.75
	1.00	**0.89**	0.80	0.33	**0.89**	0.50	0.82	0.33	**0.89**	**0.89**	0.78	0.82	0.82
OKE 2015 Task 1 gold standard sample	0.62	0.67	0.71	0.25	0.54	0.41	**0.78**	**0.77**	0.72	0.75	0.76	**0.78**	**0.78**
	0.64	0.65	0.68	0.20	0.49	0.32	0.76	0.74	0.69	0.73	0.76	**0.78**	**0.77**
	0.64	0.71	0.75	0.27	0.56	0.44	**0.81**	**0.81**	0.77	0.79	0.80	**0.82**	**0.82**
	0.64	0.67	0.72	0.22	0.53	0.35	**0.79**	0.77	0.73	0.76	0.78	**0.80**	**0.79**

Table 6

Results of NED systems reported by Gerbil. The rows in each cell report the F1@Micro, F1@Macro, InKB F1@Micro, and InKB F1@Macro respectively, in which **red** marks the highest F1 and **blue** marks the second highest F1.