Robust Entity Linking via Random Walks

Zhaochen Guo
Department of Computing Science
University of Alberta
zhaochen@ualberta.ca

Denilson Barbosa
Department of Computing Science
University of Alberta
denilson@ualberta.ca

ABSTRACT

Entity Linking is the task of assigning entities from a Knowledge Base to textual *mentions* of such entities in a document. State-of-the-art approaches rely on lexical and statistical features which are abundant for popular entities but sparse for unpopular ones, resulting in a clear bias towards popular entities and poor accuracy for less popular ones. In this work, we present a novel approach that is guided by a natural notion of *semantic similarity* which is less amenable to such bias. We adopt a unified semantic representation for entities and documents—the probability distribution obtained from a random walk on a subgraph of the knowledge base—which can overcome the feature sparsity issue that affects previous work. Our algorithm continuously updates the semantic signature of the document as mentions are disambiguated, thus focusing the search based on context. Our experimental evaluation uses well-known benchmarks and different samples of a Wikipedia-based benchmark with varying entity popularity; the results illustrate well the bias of previous methods and the superiority of our approach, especially for the less popular entities.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering, search process*

Keywords

Entity linking, relatedness measure, random walk

1. INTRODUCTION

Entity Linking (EL) is the task of assigning unique identifiers of *entities* in a Knowledge Base (KB) to *mentions* of named entities in a text document. EL is key for Information Extraction (IE) and many other applications. For instance, it enables expanding or correcting a KB with facts extracted from documents–a task called Knowledge Base Population [15]. Another application is Semantic Search, the emerging paradigm of Web search that combines Information Retrieval approaches over document corpora with KB-style query answering and reasoning to offer more accurate and concise answers to Web searches.

EL is challenging due to the inherent ambiguity of natural language: most entities can be referred to in different ways, and the same mention may refer to multiple real-world entities, as illustrated in the following examples:

Example 1. ***Saban**, previously a head coach of **NFL's Miami**, is now coaching **Crimson Tide**. His achievements include leading **LSU** to the **BCS National Championship** once and **Alabama** three times.*

Example 2. *After his departure from **Buffalo**, **Saban** returned to coach college football teams including **Miami**, **Army** and **UCF**.*

In Example 1, both *Crimson Tide* and *Alabama* refer to the same football team, the Alabama Crimson Tide of the University of Alabama. On the other hand, *Saban* refers to two different coaches: Nick Saban in Example 1 and Lou Saban in Example 2. Similarly, *Miami* in Example 1 refers to the NFL football team Miami Dolphins, while in Example 2 it refers to the college football team Miami Hurricanes.

The EL task can be cast as an all-against-all matching problem: given m mentions in a document and n entities in a KB, perform the $m \times n$ comparisons and pick the ones with the highest similarity (one for each mention, of course). This is prohibitively expensive and unnecessary, however. Most methods, including ours, operate in two stages: the first is to select a suitable set of candidate entities for each mention, and the second is to perform the actual mention disambiguation. Selecting candidate entities is done, primarily, by consulting alias dictionaries (e.g., produced from a Wikipedia corpus). In the examples above, both coaches Lou Saban and Nick Saban would be picked as candidates for the mention *Saban*, as would the companies Saban Capital Group and Saban Entertainment. As for the disambiguation phase, most methods in the literature can be divided into two main groups, discussed next.

Local Methods. The first EL systems focused mainly on lexical features, such as contextual words or named entities, surrounding each mention in the document [2,3]. They disambiguated each mention independently, typically by ranking the candidate entities according to their feature similarity with the mention, and picking the most similar candidate. These approaches work best when the context is rich

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'14, November 3–7, 2014, Shanghai, China.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2661887.

enough to uniquely identify a mention, which is not always the case. For instance, both *Saban* and *Miami* in the examples above are hard to disambiguate locally. Another issue is the *feature sparsity problem*, which arises because many mentions to the same entity are too dissimilar to match (e.g., *Edmonton* and *The City of Champions*), and are thus missed by the similarity measure. False-positives are also common: the mention to *Miami* in the examples above may mislead the disambiguation of *Saban*, leading to the erroneous conclusion that both sentences refer to a single football coach.

Global Methods. Other approaches aim at taking into account the semantics of the mentions and candidate entities, represented as a *graph* consisting of entities and links in the KB. In general, they start with a graph that has all m mentions in the document, linked to every one of their candidate entities in the KB. In turn, the candidate entities are also linked to a small subset of their full neighborhood in the KB. Disambiguation in this approach is done *collectively* on all mentions at the same time [6, 13, 16, 23]: they seek to find a forest embedded in the constructed graph in which each mention remains linked to a single candidate entity. This approach is motivated by the premise that the disambiguation of one mention contributes to the disambiguation of the remaining mentions in the same document. For example, linking *Crimson Tide* to Alabama Crimson Tide will make it easier to disambiguate *Saban* in Example 1 to Nick Saban since he is more semantically related to that team (i.e., there is a link between these entities in the KB).

Of course, the notion of semantic relatedness used by each method is key to its accuracy and cost. One successful strategy uses a text-based relatedness measure [12] based on weighted (multi-word) *keyphrases* about the entities, collected at the time the KB is built. This approach works best when the input document mentions the candidate entities in similar ways to those in the corpus used for the KB construction. Another approach uses the set of entities directly connected in the KB [20]. Doing so, however, ignores entities that are indirectly connected but semantically related, making limited use of the KB graph.

Another observation about global methods is that, to produce meaningful results, the search must be constrained so that it produces a small mention-to-entity assignment (i.e., forest in the original graph) with high global coherence (e.g., based on semantic relatedness). However, finding such an assignment is NP-Hard [16], and all methods turn to approximate algorithms or heuristics. One interesting way to cope with the complexity is to use a greedy search, starting with mentions that are easy to disambiguate (e.g., have just one candidate entity) [21].

1.1 Our approach

This paper introduces Rel-RW (Robust EL with Random Walks): a global EL approach based on an iterative mention disambiguation algorithm. We use a novel measure of semantic relatedness: we build an expanded entity graph like other global approaches, and we represent each candidate entity by the stationary probability distribution resulting from a random walk with restart [24] on that graph. We call such distributions the *semantic signature*s of the entities. As demonstrated in the personalized PageRank algorithm [11], a random walk with restart on a graph can propagate information along the edges and provide a relatedness measure between indirectly connected nodes. The probability value in the stationary distribution can be viewed as the interestingness these target entities have on each entity in the graph, with higher value indicates higher relatedness between each entity and target entities. Thus, our semantic signatures capture the semantics of the entities in terms of their relevance with respect to all other entities in the entity graph. Because the graph is built for each document, our semantic signatures are specific to the domain of the document. Semantic relatedness in our approach is measured using Zero-KL Divergence [14], although other ways are possible (e.g., cosine similarity of the semantic signatures).

Our disambiguation algorithm is iterative. Suppose that at a given round there are k mentions already disambiguated. The algorithm uses a random walk with restart using all k entities to find the *semantic signature of the document*, which is used to compute the similarity of all ambiguous mentions against their candidate entities. The algorithm picks one entity and greedily assigns it to the candidate with highest total score above a threshold, or NIL if no such candidate exists, and proceeds to the next round, re-computing the semantic signature of the document as mentions are disambiguated. For the first round, the semantic signature of the document can be initialized in one of two ways. First, if there are unambiguous mentions [21] at the beginning of the first round, those are used to define the signature. Otherwise, the candidate entities of mentions weighted by their importance and prior probability are used.

There are several advantages in using semantic signatures. First, they capture the semantics of an entity in a more fine-grained manner than the 0-1 coarse weighting in local approaches. Second, through the random walk on an entity graph, they incorporate those indirectly-connected but semantically-related entities into the representation of the target entity, which are ignored by the current link-based relatedness measures. Third, the global coherence assumption—that coherent entities form a dense subgraph—still applies on an entity graph carefully constructed from the knowledge base, which means that the less popular entities in a KB can still get high score in the entity graph since they are semantically more related with other entities in the graph thus receive more connections. As confirmed by the experiments reported in this paper, this enables our approach to excel in disambiguating less popular entities, without compromising accuracy for very popular ones. Finally, the random walk with restart can be used on a set of target entities to compute the semantic signature of documents (represented by a set of target entities). In other words, the semantic signature is a unified representation for entities and documents, and can greatly facilitate the measure of global coherence between entities and documents.

The contributions of this work are:

- We propose a unified semantic representation for entities and documents using the stationary distribution through a random walk with restart on an entity graph.
- We propose a robust entity linking algorithm using the semantic representation to improve the effectiveness.
- We experimentally evaluate our EL system on several public benchmarks, and make comparisons with 6 state-of-the-art EL systems and 2 strong baselines.
- We construct a set of datasets containing entities with varying popularity, and use them to experimentally evaluate the EL systems.

2. PROBLEM FORMULATION

Given a knowledge base (KB), usually in the form of a graph, and a document d with mentions marked up (usually through a named entity recognition process), entity linking aims at assigning unique entities from the KB to those mentions, whenever appropriate. More precisely:

DEFINITION 1 (ENTITY LINKING). *Given a set of mentions $M = \{m_1, \dots, m_n\}$ in a document d, and a knowledge base KB whose entity set is E, the problem of entity linking is to find an assignment $\Gamma : M \to E \cup \{\mathsf{NIL}\}$.*

As usual, NIL is used for mentions outside of the KB.

As mentioned, REL-RW is a global EL approach. Our goal is to find the best assignment Γ that is contextually compatible with mentions in M, and has maximum coherence. Formally, the solution to the EL problem is an assignment Γ^* maximizing the following objective function:

$$\Gamma^* = \arg\max_{\Gamma} \left(\sum_{1}^{N} \phi(m_i, e_i) + \Psi(\Gamma) \right) \tag{1}$$

in which $\phi(m_i, e_i)$ measures the local compatibility of m_i and e_i, and $\Psi(\Gamma)$ measures the global coherence of an assignment Γ.

2.1 Disambiguation with Semantic Signatures

Since the compatibility measure $\phi(m_i, e_i)$ is straightforward to compute[1], the focus of the objective function above is the measure of coherence among entities in an assignment. Recall that solving Equation 1 is NP-hard in general. Our approach is to use a greedy and iterative heuristic: in each iteration, we re-compute the semantic signature of the document d, and disambiguate a mention m according to:

$$\Gamma(m) = \arg\max_{e_i \in CS(m)} (\phi(m, e_i) + \psi(e_i, d)) \tag{2}$$

where $CS(m)$ are the candidate entities for m, and $\psi(e_i, d)$ is the semantic relatedness measure. In the next iteration, the entity e_i linked to the mention m will be taken into account when re-computing the semantic signature of the document. By doing so, we guide the search and increase the coherence of the resulting assignment.

***Linking to* NIL.** A mention is linked to NIL in one of two cases: (1) when the mention has no good candidate entities; and (2) when the similarity score of the entity maximizing Equation 2 and the document is below a threshold. Both thresholds are, of course, application-specific. In future work, we will study their impact on typical EL tasks.

3. SEMANTIC SIGNATURES

Knowledge bases are graphs where the nodes are entities and edges between entities exist whenever they are semantically related. The graph connectivity can be used to measure relatedness between entities as follows. Let $G = (V, E)$ be a graph and let $\bar{V} \subseteq V$ be a set of vertices such that $|\bar{V}| = n$. The semantic signature of a vertex $v \in \bar{V}$ is a n-dimensional vector where the weight of index i is the *relatedness* between v and the vertex i in $\bar{V}$. In this work, relatedness is defined as the probability that node i is visited in a random walk process restarting at vertex v. We call each n-dimensional vector the *semantic signature* of the vertex v.

	e_1	$\dots$	e_i	$\dots$	e_j	$\dots$	e_n
e_1	w_{11}	$\dots$	w_{1i}	$\dots$	w_{1j}	$\dots$	w_{1n}
$\vdots$							
e_i	w_{i1}	$\dots$	w_{ii}	$\dots$	w_{ij}	$\dots$	w_{in}
$\vdots$							
e_j	w_{j1}	$\dots$	w_{ji}	$\dots$	w_{jj}	$\dots$	w_{jn}
$\vdots$							
e_k	w_{k1}	$\dots$	w_{ki}	$\dots$	w_{kj}	$\dots$	w_{kn}
d	w_{d1}	$\dots$	w_{di}	$\dots$	w_{dj}	$\dots$	w_{dn}

Figure 1: Semantic signature of entities and documents.

Th notion of signature extends naturally to a *set* of k vertices of $\bar{V}$: perform a random walk with restart from the k vertices and consider the resulting probability distribution over the n vertices. Thus, we can obtain the semantic signature of an entire *document* by performing the random walks from the entities that are mentioned in it.

Figure 1 illustrates the idea of the semantic signature of entities and documents.

3.1 Random Walk with Restart

A random walk with restart is a stochastic process to traverse a graph, resulting in a probability distribution over the vertices corresponding to the likelihood those vertices are visited. This probability can be interpreted as the relatedness between nodes in the graph. The random walk starts with an initial distribution over the nodes in the graph, propagating the distribution to adjacent vertices proportionally, until convergence.

Let A be the transition matrix of the KB graph, with A_{ij} being the probability of reaching entity e_j from entity e_i, which can be computed as follows:

$$A_{ij} = \frac{w_{ij}}{\sum_{e_k \in OUT(e_i)} w_{ik}}$$

in which $OUT(e_i)$ is the set of entities directly reachable from e_i, and w_{ij} is the weight of the edge between e_i and e_j, defined as the number of their co-occurrences in the knowledge base (see below).

Let r^t be the probability distribution at iteration t, and r_i^t be the value of entity e_i, then r_i^{t+1} is computed as follows:

$$r_i^{t+1} = \sum_{e_j \in IN(e_i)} r_j^t * A_{ji} \tag{3}$$

in which $IN(e_i)$ is the set of entities linking to e_i.

As customary, we incorporate a random restart probability in the *preference vector* to avoid the issues caused by sinks and guarantee convergence. Formally, the random walk model can be modeled as:

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

where $\vec{v}$ is the preference vector, and $\sum v_i = 1$. We also follow the standard convention and set $\alpha = 0.85$.

When a random walk process converges to a stationary state, we obtain a *stationary distribution*, which is what we use as our semantic signature.

[1] We combine the prior probability and context similarity to measure the local compatibility in our system.

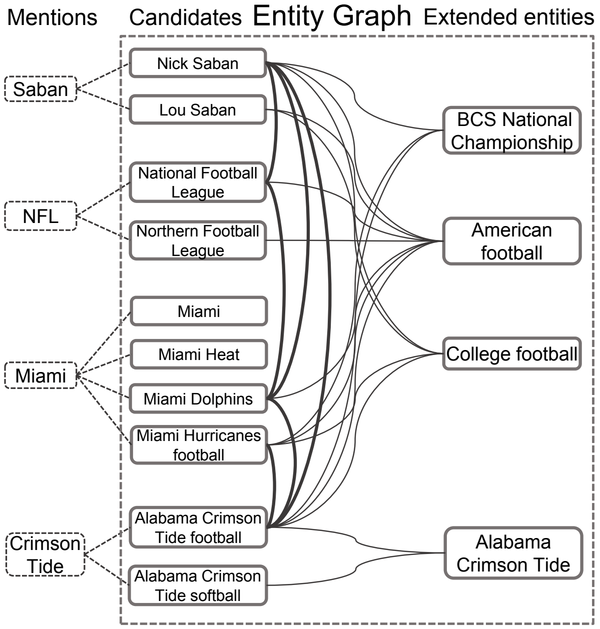

Figure 2: A mini entity graph.

3.2 Entity Graph Construction

We follow the standard practice of deriving a KB from Wikipedia. The entity graph $G = (V, E)$ is such that V contains the entities (i.e., individual articles in Wikipedia) and the edges in E are added if the two entities: (1) are mentioned in the same Wikipedia article, within a window [4] of 500 words[2], or (2) there is an explicit hyper-link from one entity to the other. In this way, we construct a very dense entity graph consisting of over 4 million entities (i.e., Wikipedia articles). However, with respect to the semantic signature computation, the efficiency on this large graph will be a big concern. Thus, we seek to construct a much smaller subgraph that can improve the efficiency without sacrificing the effectiveness.

Subgraph Construction. The subgraph construction starts with a set of candidate entities for mentions in a document. Given a set of mentions M, the first step is to find good candidate entities from the KB—a set of entities that could be referred to by mentions in M (details about candidate selection will be described in Section 4.1). The candidate entities and the edges among them form an initial entity graph, which is expanded by adding all entities adjacent to a candidate entity (and the edges within that subgraph too). Figure 2 illustrates the process: the leftmost column shows the mentions, the middle column lists the set of candidate entities, while the rightmost column has the extra entities added to the graph.

We post-process the expanded graph to remove noisy entities and reduce the size of the subgraph by pruning non-candidate entities that are connected to just one candidate entity as well as entities with low degree (200 was experimentally chosen as the minimum value). Our KB graph is so dense that, without pruning, the expanded subgraph of most mentions quickly becomes prohibitively large. Notice that candidate entities are never pruned, to make sure unpopular entities are not ignored.

Discussion. In addition to reducing the graph size and improving the efficiency of the random walk process, another advantage of the smaller subgraph we build is that candidate entities and their semantically-related entities are more densely connected (as per co-occurrence and direct linkage in Wikipedia), which preserves the global coherence assumption in most global approaches. As illustrated in Figure 2, the football related entities such as *Nick Saban*, *National Football League*, and others form a much denser subgraph than other entities.

3.3 Semantic Signature Computation

With the entity graph, we can then compute the semantic signatures of entities with the random walk model.

3.3.1 Semantic Signature of an Entity

To compute the semantic signature for a target entity e_i, we need to ensure that the random walk always restarts from e_i. This can be done by setting the preference vector $\vec{v}$ with $v_i = 1$, and $v_{j(j \neq i)} = 0$.

3.3.2 Semantic Signature of a Document

In principle, computing the semantic signature of a document is no different than doing so for a single entity. Given a set of entities E_d representing a document, we set their preference probability in vector $\vec{v}$, and then compute the semantic signature of the document through the random walk with restart from entities in E_d.

However, there are two problems here. First, we do not know the true entity set E_d that represents the document (finding this set is the goal of EL after all). Second, it is not clear how to set the weights in the preference vector. Different entities may have different importance to the document and a uniform weight may not reflect their importance. To solve these two problems, we adopt the following strategies.

Finding E_d. We say a mention is *unambiguous* if there is only one entity in the KB associated with it. Unambiguous mentions have been shown to help in the EL task [21]. In Example 1, *BCS National Championship* is one such unambiguous mention, which is very useful to disambiguate other mentions. We initialize the set E_d with the referent entities of all unambiguous mentions in M, and expand it as more mentions are disambiguated.

In case all mentions in the document are ambiguous, we approximate E_d using candidate entities of the mentions in M. Although this approximation can bring in a lot of noise, and thus decrease the accuracy of the disambiguation, the effectiveness may not be affected much by the noisy entities when the true entities are well connected in the graph.

Determining Weights. The preference probability of an entity e_i could be affected mainly by two factors: $P(e_i, m)$—the probability mention m refers to e_i, and $I(m)$—the importance of the mention in its document.

For the case that the referent entities of unambiguous mentions are used, the $P(e_i, m)$ is 1 since e_i is considered as the true entity of m. When using the candidate enti-

[2] 500 was chosen based on intuition, without detailed experimental evaluation. Tuning the best window size will be a subject of future work.

ties, the probability $P(e_i, m)$ can be measured in several ways. Prior probability is a statistical measure shown to be a strong baseline [8]. Other alternatives are using the context similarity between e_i and m or assigning weights uniformly. We experimented with several options, and, fortunately, as shown in our experimental evaluation, Rel-RW is very robust to the choice of weights, consistently yielding good results.

The importance of a mention $I(m)$ is measured using the standard *tf-idf* scheme.

Combining the two factors $P(e_i, m)$ and $I(m)$ together, we can compute the preference probability as follows:

$$\vec{v}_i = I(m) * P(e_i, m) \tag{5}$$

With the preference vector $\vec{v}$, the semantic signature of a document can be computed using a random walk with restart on the entity graph. As shown in Figure 1, the row for document d gives its semantic signature.

3.3.3 Computational Cost

Suppose there are total K candidates for all mentions in M, we need to compute $K + 1$ semantic signatures (K for entities, and 1 for the document). To improve the efficiency, we adopt the following methods. First, we rank candidates by their relevance to mentions and select only the top candidates for each mention. Second, we limit the size of the entity graph by pruning uncommon entities as described above. Third, we parallelize the computation of the semantic signatures for the candidate entities. Our current implementation of Rel-RW was done with accuracy in mind, and we have not yet investigated how to properly engineer it to make it real-time. Nevertheless, it is competitive with other off-line state of the art methods. Future work will explore the rich literature on speeding up the computation of random walks, as well as efficient indexing mechanisms for the efficient computation of the entity graphs from the KB.

4. MENTION DISAMBIGUATION

After computing the semantic signatures of the document and entities, the next step is to disambiguate mentions in the document.

4.1 Candidate Generation

Mention disambiguation starts with generating candidates for each mention in M from the knowledge base. For this purpose, we use an alias dictionary collected from Wikipedia titles, redirect pages, disambiguation pages, and the anchor text in Wikilinks [6]. The alias dictionary maps each alias to a list of entities it refers to in Wikipedia.

Given a mention m, we search its name against the alias dictionary and select the set of entities the mention name refers to. This would generate a long list of candidates. To keep the list short and improve the efficiency, we prune noisy candidates according to the following two criteria.

- Prior probability. We rank entities by their prior probability $P(e_i|m)$, and select the top x entities as the candidates to keep the popular candidates ($x = 10$ was used in the experiments reported here).
- Context similarity. We select the top y entities ranked by their context similarity with the mention to increase the recall of candidate generation (again, $y = 10$ in the experiments reported).

Algorithm 1 Iterative Mention Disambiguation

Input: $M = \{m_1, m_2, \ldots, m_n\}$, $G = (V, E)$
Output: An assignment $\Gamma = \{e_1, e_2, \ldots, e_n\}$, in which e_i is the referent entity of m_i

```
$E_d = \emptyset, \Gamma = \emptyset$
for all $m_i \in M$ do
    if $|CS(m_i)| = 0$ then
        $\Gamma(m_i) = NIL$
    else if $|CS(m_i)| = 1$ then
        $E_d = E_d \cup CS(m_i)$; $\Gamma(m_i) = e^* \in CS(m_i)$.
    end if
end for

if $|E_d| = 0$ then
    Initialize $E_d$ with candidates of mentions in $M$.
end if

$SS(d) = ComputeSignature(G, E_d)$
Rank $m_i \in M$ by their ambiguity $|CS(m_i)|$.

for all $m_i \in M$ and $|CS(m_i)| > 1$ do
    for $e_j \in CS(m_i)$ do
        $SS(e_j) = ComputeSignature(G, e_j)$
        $\psi(e_j, d) = SemanticSimilarity(SS(e_j), SS(d))$
        $SimilarityScore(e_j) = \phi(m_i, e_j) + \psi(e_j, d)$.
    end for

    Rank candidates $CS(m_i)$.
    Select the target entity $e^*$ with the maximum score.
    Remove the rest candidates from $CS(m_i)$.

    if $SimilarityScore(e^*) < \rho$ then
        $\Gamma(m_i) = NIL$; Remove $e^*$ from $CS(m_i)$.
    else
        $E_d = E_d \cup \{e^*\}$; $\Gamma(m_i) = e^*$.
    end if

    if $E_d$ is changed then
        $SS(d) = ComputeSignature(G, E_d)$
    end if
end for
```

4.2 Mention Disambiguation

The second step is to select the referent entity from the candidate set $CS(m)$. As defined in the objective function 2, the referent entity is the entity that is most coherent with the mention and the document.

4.2.1 Semantic Relatedness

Let $SS(e_i)$ be the semantic signature of a candidate entity $e_i \in CS(m)$, and $SS(d)$ be the semantic signature of the document d. There are several ways one can compare these probability distributions to estimate the semantic similarity of the m and d. One standard way of doing so is to use The Kullback-Leibler (KL) divergence: given two probability distributions P and Q, their KL divergence measures their distance, as follows:

$$D_{KL}(P \parallel Q) = \sum_i P_i \log \frac{P_i}{Q_i} \tag{6}$$

In this work, we use Zero-KL Divergence [14], a better approximation of the KL divergence that handles the case

	Datasets								
	MSNBC			AQUAINT			ACE2004		
Systems	**Accuracy**	**F1@MI**	**F1@MA**	**Accuracy**	**F1@MI**	**F1@MA**	**Accuracy**	**F1@MI**	**F1@MA**
PriorProb	85.98	86.50	87.15	84.87	87.27	87.16	84.82	85.49	87.13
Local	77.43	77.91	72.30	66.44	68.32	68.09	61.48	61.96	56.95
Cucerzan	87.80	88.34	87.76	76.62	78.67	78.22	78.99	79.30	78.22
M&W	68.45	78.43	80.37	79.92	85.13	84.84	75.54	81.29	84.25
Han'11	87.65	88.46	87.93	77.16	79.46	78.80	72.76	73.48	66.80
AIDA	76.83	78.81	76.26	52.54	56.47	56.46	77.04	80.49	84.13
GLOW	65.55	75.37	77.33	75.65	83.14	82.97	75.49	81.91	83.18
RI	88.57	90.22	90.87	85.01	87.72	87.74	82.35	86.60	87.13
REL-RW	**90.12**	**91.37**	**91.73**	**88.99**	**90.74**	**90.58**	**86.93**	**87.68**	**89.23**

Table 1: Effectiveness of various EL systems on the MSNBC, AQUAINT, and ACE2004 datasets.

when Q_i is zero.

$$ZKL_\gamma(P,Q) = \sum_i P_i \begin{cases} \log \frac{P_i}{Q_i} & Q_i \neq 0 \\ \gamma & Q_i = 0 \end{cases} \qquad (7)$$

in which γ is a real number coefficient. Following the recommendation in [14], we set $\gamma = 20$, arriving at the semantic similarity used in Equation 2:

$$\psi(e_i, d) = \frac{1}{ZKL_\gamma(SS(e_i), SS(d))} \qquad (8)$$

4.2.2 Iterative Disambiguation Algorithm

With the candidate set and semantic relatedness measure, we can then disambiguate mentions by selecting entities maximizing the ranking function in Equation 2.

As described in Section 3, the representative entities are crucial for computing the semantic signature of a document. Using candidates of mentions to represent a document may bring in noisy entities that are not useful. While unambiguous mentions are informative, they could be rare so that only partial semantics of the document is captured. To address this issue, we propose an iterative algorithm which utilizes the disambiguation results of previous iterations to progressively update the representative entity set and the semantic signature of the document.

Instead of ranking candidates and choosing entities with the highest score for each mention independently, our algorithm performs the disambiguation iteratively in the order of their ambiguity (which is measured by their number of candidates) from the least ambiguous mention to the most ambiguous one. In each iteration, the results of previous iterations are used to update the representative entity set and the semantic signature of the document. For the case we start with the candidates of mentions (when no unambiguous mentions exist), the representative entity set is changed to the disambiguation results right after they become available.

Algorithm 1 shows the disambiguation process. We first disambiguate mentions with none or one candidate and initialize the entity set E_d with the entities of unambiguous mentions (Lines 2-8). In case all mentions are ambiguous, the candidates of mentions are used to initialize E_d (Lines 9-11). We then compute the semantic signature of the document using E_d (Line 12), and sort mentions according to their ambiguity (Line 13). In each iteration of the disambiguation process, we update the similarity score between each candidate e_i and document d with the latest semantic signature (Lines 15-19). Note that the semantic signatures of all candidate entities are pre-computed. After that, we rank the candidates and choose the entity with the maximum score (Lines 20-22). If the score of the best candidate for a mention is below a threshold ρ, we link the mention to NIL (Line 24). Otherwise, we update the entity set E_d using the linked entity e^* (Line 26), and the semantic signature of the document with the enriched E_d. The new $SS(d)$ will be used for disambiguation in the next iteration (Line 16). The whole process continues until all mentions are disambiguated.

5. EXPERIMENTAL VALIDATION

We now report on a comprehensive experimental evaluation of our REL-RW method, comparing it to the state-of-the-arts and strong baselines.

Baselines. We selectively tested 6 representative competitor systems: *Cucerzan* [6]—the first collective EL system that solved the EL as an optimization problem, M&W [21]—a representative machine learning EL system, Han'11 [10]—a graph-based collective system exploiting the random walk model on an entity graph to jointly link mentions, AIDA [13]—a collective approach tackling entity linking as a dense subgraph problem, GLOW [23]—a system combining local and global features for entity linking, and RI [5]—the start-of-the art EL system using relational inference for mention disambiguation.

In addition, we also test 2 strong baselines: *PriorProb* which links mentions to the entities with the highest prior probability, and *Local* which chooses the candidate with maximum local compatibility $\phi(e_i, m)$.

Datasets. We use 3 well-known public benchmarks: (1) MSNBC [6], with 20 news articles from 10 different topics (2 articles per topic) and 739 mentions in total; (2) AQUAINT, compiled by Milne and Witten [21], with 50 documents and 727 mentions from a news corpus from the Xinhua News Service, the New York Times, and the Associated Press; and (3) ACE2004 [23], a subset of the ACE2004 Coreference documents with 57 articles and 306 mentions, annotated through *crowdsourcing*.

Evaluation Measures. Given the ground truth T and output of EL systems O, in which T_{ent} and O_{ent} are the sets of mentions linking to entities, and T_{nil} and O_{nil} are the sets

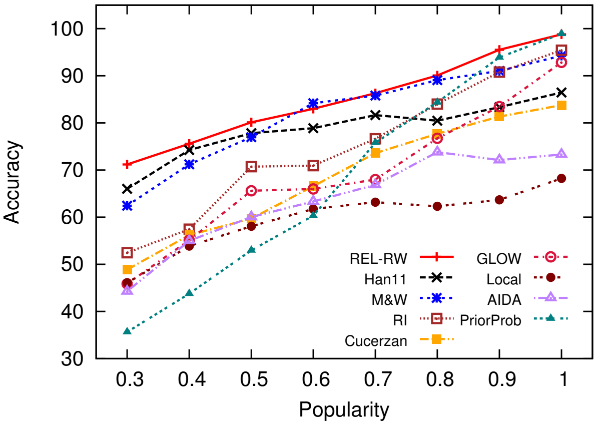

Figure 3: Accuracy vs entity popularity.

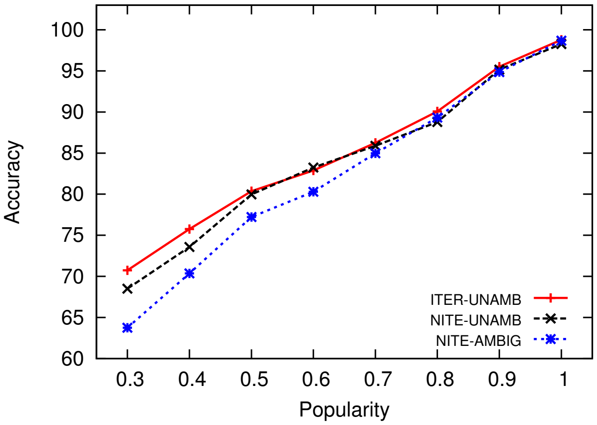

Figure 4: Accuracy of Rel-RW with different configurations.

of mentions linking to NIL respectively ($T = T_{ent} \cup T_{nil}$, $O = O_{ent} \cup O_{nil}$, and $|T| = |O|$), we use the standard *accuracy, precision, recall*, and *F1*:

$$\begin{aligned}
\mathit{Accuracy} &= \frac{|T_{ent} \cap O_{ent}| + |T_{nil} \cap O_{nil}|}{|T|} \\
\mathit{Precision} &= \frac{|T_{ent} \cap O_{ent}|}{|O_{ent}|} \\
\mathit{Recall} &= \frac{|T_{ent} \cap O_{ent}|}{|T_{ent}|} \\
F1 &= \frac{2 * \mathit{Precision} * \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}}
\end{aligned}$$

5.1 Results on Public Benchmarks

Table 1 lists the results of these EL systems on the 3 benchmarks in terms of accuracy and F1, aggregated across mentions (micro-averaged, indicated as **F1@MI**) and across documents (macro-averaged, **F1@MA**).

Several observations are worth noting here. First, the *Local* baseline indicates that text-based features alone cannot solve the EL problem satisfactorily. Combining local compatibility with the semantic relatedness, as in *Han'11*, provides substantial gains. The *RI* system, which takes advantage of the relational constraints, performs much better than most systems. Our system REL-RW outperforms all the state-of-the-arts and the baselines on the three datasets, showing the superiority of our semantic signatures and the mention disambiguation algorithm. It also shows that the coherence between entities and the document may be a better choice for the measure of global coherence.

Another general observation is that there is quite a bit of variability in the relative effectiveness of the systems across benchmarks. The exceptions to this rule are the *RI* and REL-RW as well as the *PriorProb* baseline. In fact, *PriorProb* consistently outperforms many EL systems. This points to limitations in the benchmarks themselves—they are clearly biased towards popular entities, and thus, not representative of all scenarios where EL is necessary.

5.2 Accuracy vs Popularity

In order to investigate how the popularity (measured by the prior probability) of entities affects the effectiveness of the competing systems, we devised the following experiment. Ideally, we would use multiple subsets from the Wikipedia dump, where all entities have the same popularity. However, such datasets are very hard to obtain. Instead, we used the *PriorProb* baseline as a proxy: we applied that baseline on all articles, and grouped them by the resulting accuracy, aggregating at one decimal place. For example, documents whose accuracy are in the range $[0.3, 0.4)$ are grouped together. We randomly chose 40 articles from each group, restricting selections to articles with 20 to 40 Wikilinks. We also resolved the redirect entities and removed mentions whose referent entities did not exist any more. In this way, we obtained 8 datasets, each of which, on average, had about 1000 mentions to be disambiguated.

Figure 3 compares all systems and baselines on the datasets we produced. As designed, the *PriorProb* shows a linear increase in accuracy as the "aggregate popularity" of mentions in the datasets increases. It is worth mentioning that this experiment is particularly favorable to the M&W system, which is trained on Wikipedia itself[3]. Thus, these results must be taken in consideration with those on the other benchmarks (Table 1). A general trend observed for all systems is that their accuracy improves as more popular entities are used. However, our REL-RW is the only system that is consistently superior to the *PriorProb* baseline. This somewhat surprising result indicates that there is quite a lot of room for improvement before EL can be successfully applied to the "long-tail" of the Web, given their bias towards popular entities.

The high accuracy of REL-RW on unpopular mentions is partially explained by the way we construct our entity graph. As described in Section 3, the graph is initialized with candidates, and expanded using their neighbors to increase connectivity. In this way, semantically-related entities are likely to form a dense subgraph. This also works for unpopular entities, on which the lexical and statistical features used by other approaches fail. Another factor is the semantic relatedness between indirectly connected entities which can be measured by the semantic propagation in the random walks. This effect can also be verified in the *Han'11* system,

[3] For the sake of validity, we ensured that none of the articles in our tests appear in their training data.

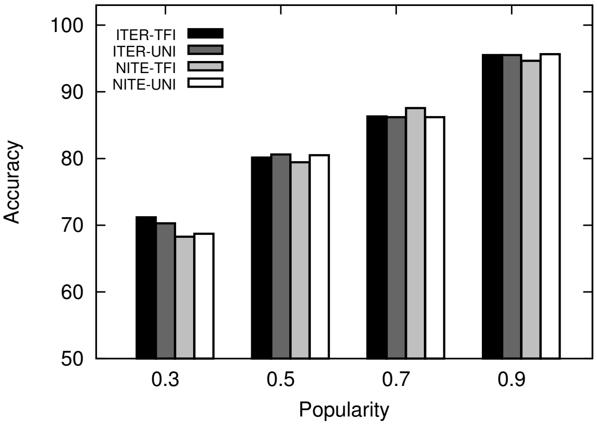

Figure 5: Accuracy of Rel-RW using unambiguous mentions with different weighting schemes.

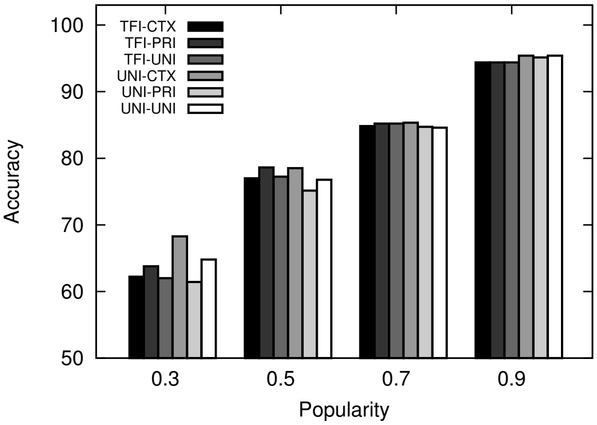

Figure 6: Accuracy of Rel-RW using candidates with different weighting schemes.

which utilizes the random walk for semantic relatedness and outperforms most systems on unpopular entities.

5.3 Evaluation on the 2011 TAC Entity Linking Task

We also performed an evaluation of Rel-RW on the TAC 2011[4] Entity Linking task. Compared to the previously discussed benchmarks, the TAC dataset contains many more abbreviations and acronyms, making the mentions more ambiguous. We perform a query expansion on the abbreviation mentions by extracting definitions of abbreviations using patterns *definition (Abbrev)* and *Abbrev (definition)*.

Table 2 shows the results of our Rel-RW system, *RI* and a few top EL systems in the submissions, in terms of linking accuracy. Rel-RW is very competitive overall, virtually tying with RI and the top submissions to the TAC contest. It is worth noting that the MS-MLI system exploits external web search logs for candidate generation and additional training datasets.

5.4 Sensitivity To Parameters

System	Accuracy
LCC	86.1
MS-MLI	86.8
RI	86.1
NUSchime	86.3
REL-RW	86.4

Table 2: Accuracy of EL systems on 2011 TAC EL task.

The following experiments are aimed at determining how robust Rel-RW is relative to the choice of representative entities and weighting schemes. We compare the iterative process with the non-iterative process using two different entity sets and different weighting schemes on the Wikipedia-based benchmarks. For entities, we use the referent entities of unambiguous mentions or the candidates of mentions. For weighting scheme, we use *tf-idf* and *uniform* for the importance of mentions $I(m)$, and *uniform*, *prior probability*, and *context similarity* to weight the mention-entity compatibility. For simplicity, we use unambiguous mentions to refer to the referent entity of unambiguous mentions.

Representative entities. We first focus on the disambiguation process and representative entities. Figure 4 shows the results with three different configurations: iterative process with unambiguous mentions (ITER-UNAMB), non-iterative process with unambiguous mentions (NITE-UNAMB), and non-iterative process with candidates of mentions (NITE-AMBIG). The results are reported using the average value of different weighting schemes. Note that we do not report the results of iterative process on ambiguous mentions, since we change the entity set of candidates to the disambiguation results after the first iteration, which is the same as the iterative process on unambiguous mentions.

As we see from Figure 4, the effectiveness of the iterative process is better than the non-iterative process. This is because the iterative process exploits the entity linking results in previous iterations to enrich the representative entity set which can improve the semantic representation of documents and the overall effectiveness while the non-iterative process only uses the entities of unambiguous mentions. We also found that the entity set of unambiguous mentions was a much better approximation of documents than the candidates of mentions.

Another observation is that the accuracy gap among different configurations gradually decreases as the popularity of entities in datasets increases. The explanation could be that in datasets with low popularity, the true entity of a mention is rarely the candidate with the highest prior probability or context similarity, so that the semantic similarity plays a more important role for the disambiguation. While in datasets with high popularity, the difference is much smaller.

Weighting schemes. Figure 5 and Figure 6 show the results of various weighting schemes using unambiguous mentions and candidates of mentions respectively. As we can see from the figures, different weighting schemes make almost no difference on the overall effectiveness of the EL system, which means that the semantic signature of a document is more dependent on the representative entities than the weighting of each entity. It also means that the entity graph

[4] http://www.nist.gov/tac/2011/

structure is more important for the semantic signature than the weighting schemes.

Discussion. The results of these experiments indicate that our method is fairly robust as the construction of the entity graph requires essentially no parameter tuning.

6. RELATED WORK

Earlier work on Entity Linking disambiguated each mention in isolation using a compatibility function to approximate the likelihood of an entity being the referent entity for a mention, and treated entity linking as a ranking problem which chose the candidate with the highest compatibility. In these approaches, mentions and entities are represented as feature vectors and vector similarity measures are used to estimate their compatibility. The most common local features include lexical features such as bag-of-words or named entities from surrounding context and statistical features such as the prior probability of entities given a mention from a knowledge base. Unsupervised approaches [2, 3] commonly use the cosine similarity of feature vectors to measure the compatibility, while supervised approaches [7, 18, 21, 25–27] exploit various classifiers trained on labeled datasets (often derived from Wikipedia) to predict the compatibility. By restricting themselves to local features, these methods suffer from the data sparsity problem. Moreover, individually linking mentions does not take into account the inherent semantic coherence among them.

More recent EL systems follow the global coherence hypothesis and take into account the semantic relations between mentions and entities, employing various measures of semantic relatedness. Most of these approaches assume that mentions in a document are semantically coherent around the subject of the document, and thus cast the entity linking as an optimization problem aiming at finding the assignment with maximum semantic coherence. Cucerzan [6] measures the global coherence using Wikipedia categories. Milne and Witten (M&W) [21] use directly connected entities to represent each entity and measure relatedness using normalized Google distance. Kulkarni et al. [16] also use the M&W semantic relatedness measure and formalize the problem as an integer linear programming problem to find the assignment collectively. Ratinov et al. [23] add the PMI (Pointwise Mutual Information) of entities into their SVM classifier for entity linking. To use more fine-grained semantics such as relations between mentions, Cheng and Roth [5] formalize the EL as an integer linear programming problem with relations as constraints, and find the assignment to meet the relational constraints. Cai et al. [4] measure the semantic relatedness using the co-occurrence of entities from a link-enriched Wikipedia corpus. AIDA [13] formalizes the EL problem as a dense subgraph problem, which aims to find a dense subgraph from a mention-entity graph such that the subgraph contains all mentions and one mention-entity edge for each mention according to their definition of graph density. Han and Sun [9] combine the local compatibility and global coherence using a generative entity-topic model to infer the underlying referent entities. Also through a generative graphical model, Li et al. [17] propose to mine additional information for entities from external corpus, which can help improve the effectiveness of entity linking, especially for entities with rare information available in the knowledge base.

The approach closest to ours is that of Han et. al [10], which uses a random walk with restart to obtain a vector of relevance for all candidates of mentions, and considers the relevance value in the vector to be the relatedness between a mention and its candidate. Like other semantic relatedness measures, their measure can only compute the relatedness between two entities. Instead, we use a unified semantic representation for both documents and entities. As a result, we can measure the coherence between entities and between entities and documents in a unified way. Also our representation can capture the semantics of unpopular entities, which makes our EL approach more robust for datasets with less popular entities. The idea of using random walk with restart has been applied on graphs constructed from the WordNet [19], with the stationary distribution to represent the semantics of words. It has been shown to be effective in the word similarity measurement [1, 14], and word sense disambiguation [22]. However, we are not aware of any previous work using the stationary distribution from random walk with restart to represent entities and documents in entity linking.

7. CONCLUSION

Entity linking mainly involves measuring the compatibility and semantic relatedness between mentions and entities, for which the semantic representation plays a critical role. The lexical and statistical features used in traditional local approaches are limited by the feature sparsity issue and cannot capture the semantics of entities. Recently proposed keyphrase and link-based representations provide richer feature sets for popular entities, but are still challenged for less popular ones. In this work, we proposed the use of the probability distribution resulting from a random walk with restart over a suitable entity graph to represent the semantics of entities and documents in a unified way. Our semantic representation uses all relevant entities from the knowledge base as features, thus reducing the effect of feature sparsity. Also the random walk with restart helps capture the semantics of unpopular entities from the entity graph.

Our experimental evaluation compared our method to 6 leading competitor systems and 2 very strong baselines, revealing the superiority and robustness of our entity linking system in a variety of settings, including 4 public benchmarks. Our method was particularly stronger when disambiguating unpopular entities, making it a good candidate to address the "long tail" in Information Extraction.

Future work. A number of opportunities for future work exist. First, several disambiguation mistakes are still possible. For instance, when we have two *Miami*s in the same sentence, one referring to the city in Florida and the other to *Miami Dolphins*, our method might mistakenly link both to the same entity. In this case, simple type information like *location* and *sports team*, which are provided by named entity recognition systems can help with the disambiguation. Combining our current approach with typing information or other kinds of constraints would help advance the field.

Our system performs multiple random walk computations, making it time consuming if implemented naively, as in our current implementation. On a standard entry-level server, the average time to disambiguate a document in our benchmarks (in the order of tens of mentions) is in the order of a few minutes. Therefore, designing proper system infrastructure with the appropriate indexes and/or parallel computing infrastructure to optimize these computations would be interesting. Moreover, other state-of-the-art systems perform other expensive operations as well, such as accessing the Web. Thus, designing objective and fair benchmarks for comparing these different approaches in a more holistic way would be of great value.

Finally, our approach, like most other systems, has many application-specific parameters (recall Section 2) and depends on specific similarity measures (e.g., to filter candidate entities). Further studies are needed to understand how the accuracy of our approach is affected by the choice of similarity measures and configuration of parameters.

Acknowledgements

This work was supported in part by grants from the Natural Sciences and Engineering Council of Canada, through its Business Intelligence Network, and Alberta Innovates Technology Futures.

8. REFERENCES

[1] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pașca, and A. Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *HLT-NAACL*, pages 19–27, 2009.

[2] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *COLING-ACL*, pages 79–85, 1998.

[3] R. C. Bunescu and M. Pașca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, 2006.

[4] Z. Cai, K. Zhao, K. Q. Zhu, and H. Wang. Wikification via link co-occurrence. In *CIKM*, pages 1087–1096, 2013.

[5] X. Cheng and D. Roth. Relational inference for wikification. In *EMNLP*, pages 1787–1796, 2013.

[6] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716, 2007.

[7] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *COLING*, pages 277–285, 2010.

[8] Z. Guo and D. Barbosa. Entity linking with a unified semantic representation. In *WWW (Companion Volume)*, pages 1305–1310, 2014.

[9] X. Han and L. Sun. An entity-topic model for entity linking. In *EMNLP-CoNLL*, pages 105–115, 2012.

[10] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *SIGIR*, pages 765–774, 2011.

[11] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.*, 15(4):784–796, 2003.

[12] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *CIKM*, pages 545–554, 2012.

[13] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *EMNLP*, pages 782–792, 2011.

[14] T. Hughes and D. Ramage. Lexical semantic relatedness with random graph walks. In *EMNLP-CoNLL*, pages 581–589, 2007.

[15] H. Ji and R. Grishman. Knowledge base population: Successful approaches and challenges. In *ACL*, pages 1148–1158, 2011.

[16] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD*, pages 457–466, 2009.

[17] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan. Mining evidences for named entity disambiguation. In *KDD*, pages 1070–1078, 2013.

[18] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, pages 233–242, 2007.

[19] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[20] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI, 2008)*, 2008.

[21] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, pages 509–518, 2008.

[22] M. T. Pilehvar, D. Jurgens, and R. Navigli. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *ACL*, pages 1341–1351, 2013.

[23] L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL*, pages 1375–1384, 2011.

[24] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, pages 613–622, 2006.

[25] W. Zhang, Y. C. Sim, J. Su, and C. L. Tan. Entity linking with effective acronym expansion, instance selection, and topic modeling. In *IJCAI*, pages 1909–1914, 2011.

[26] W. Zhang, J. Su, C. L. Tan, and W. Wang. Entity linking leveraging automatically generated annotation. In *COLING*, pages 1290–1298, 2010.

[27] Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and S. Gaffney. Resolving surface forms to wikipedia topics. In *COLING*, pages 1335–1343, 2010.