

Entity Disambiguation with Linkless Knowledge Bases

Yang Li^{1*}, Shulong Tan¹, Huan Sun¹, Jiawei Han², Dan Roth², Xifeng Yan¹
{yangli, shulongtan, huansun, xyan}@cs.ucsb.edu, {hanj, danr}@cs.uiuc.edu

¹Department of Computer Science
University of California, Santa Barbara
CA, 93106, USA

²Department of Computer Science
University of Illinois at Urbana-Champaign
IL, 61801, USA

ABSTRACT

Named Entity Disambiguation is the task of disambiguating named entity mentions in natural language text and link them to their corresponding entries in a reference knowledge base (e.g. Wikipedia). Such disambiguation can help add semantics to plain text and distinguish homonymous entities. Previous research has tackled this problem by making use of two types of context-aware features derived from the reference knowledge base, namely, the context similarity and the semantic relatedness. Both features heavily rely on the cross-document hyperlinks within the knowledge base: the semantic relatedness feature is directly measured via those hyperlinks, while the context similarity feature implicitly makes use of those hyperlinks to expand entity candidates' descriptions and then compares them against the query context. Unfortunately, cross-document hyperlinks are rarely available in many closed domain knowledge bases and it is very expensive to manually add such links. Therefore few algorithms can work well on linkless knowledge bases. In this work, we propose the challenging Named Entity Disambiguation with Linkless Knowledge Bases (LNED) problem and tackle it by leveraging the useful disambiguation evidences scattered across the reference knowledge base. We propose a generative model to automatically mine such evidences out of noisy information. The mined evidences can mimic the role of the missing links and help boost the LNED performance. Experimental results show that our proposed method substantially improves the disambiguation accuracy over the baseline approaches.

Keywords

Entity Disambiguation; Linkless Knowledge Bases; Evidence Mining; Generative Model

1. INTRODUCTION

An important component in constructing information networks is *named entity disambiguation* (NED). Given the named entity mentions extracted from unstructured text data, the goal of NED is to map them to their correspond-

*Currently at Google Inc.; this work was completed before the change in affiliation.

ing real world entities in a reference knowledge base such as Wikipedia. NED has many key applications in text analysis and understanding, e.g., tweet tagging, text classification, and ad placement. Due to natural language's inherent ambiguities, the NED task is quite challenging. The same textual mention can represent multiple different real world entities depending on the context of its appearance. For example, "Eclipse" can refer to the Java development platform, the car designed by Mitsubishi or even a breath freshener brand.

NED requires a reference knowledge base to serve as the real world entity collections to which the named entity mentions will be resolved. Usually the reference knowledge base is comprised of a set of documents with each document describing one specific entity. Almost all previous research on NED uses Wikipedia as the reference knowledge base. Despite the fact that Wikipedia covers millions of entities, most not well known or domain specific entities are not captured by Wikipedia. Therefore in closed domains (e.g. biomedicine, entertainment, enterprise, etc.), in order to achieve good NED performance, we have to rely on some domain specific knowledge bases (e.g. product catalog [21], restaurant directory [8], gene descriptions [20], etc.) as the reference knowledge bases.

The [Mitsubishi Eclipse](#) is a [sport compact](#) car that was in production between 1989 and 2011. A [convertible](#) body style was added for the 1996 model year. It was named after an unbeaten [18th-century English racehorse](#) which won 26 races, and has also been sold as the [Eagle Talon](#) and the [Plymouth Laser captive imports](#) through [Mitsubishi Motors](#)' close relationship with the [Chrysler Corporation](#). Their partnership was known as [Diamond-Star Motors](#), or DSM, and the vehicle trio through the close of the second-generation line were sometimes referred to by the DSM moniker among enthusiast circles. In Japan, it was sold at a specific retail chain called [Car Plaza](#).

Figure 1: Snapshot of hyperlinks in the Wikipedia page of *Mitsubishi Eclipse*

One particular difference between Wikipedia and closed domain knowledge bases is the existence of cross-document hyperlinks. As the largest publicly available encyclopedia in the world, Wikipedia contains not only the description pages for millions of entities, but also the huge amount of cross-document hyperlinks connecting those entities. Those links are created by 20,694,972 [1] contributors from all around the world. Figure 1 shows a snapshot of such links in the Wikipedia page of entity *Mitsubishi Eclipse*. As we can see, 9 hyperlinks (denoted with underlines) are created to connect *Mitsubishi Eclipse* with 9 different entities in Wikipedia. On the contrary, we notice that most closed domain knowledge bases contain few cross-document hyperlinks. For exam-

ple, Amazon’s product catalog is linkless. Each product has some detailed product description, however, these descriptions are not cross-linked. In order to achieve social promotion (e.g. linking product mentioned in tweets to its page in Amazon), a system should be able to disambiguate mentions to the linkless product catalog. The repository of crime cases in California Police Departments is also linkless. There each person has a profile page, but no links are provided to connect different people. In order to identify suspects (e.g. linking people mentioned in witness reports to his/her profile page), we also need to disambiguate mentions to the linkless profile repository. As closed domain knowledge bases are usually created by a limited number of domain experts, it is very costly to manually cross link all the entity mentions.

Previous research [7, 9, 12, 13, 15, 17, 23, 24, 26, 29] has tackled the NED problem (with respect to Wikipedia) by making use of various textual and structural information from Wikipedia. Generally speaking, two kinds of context-aware features are explored by those methods. The first one is *context similarity*. It defines similarity measures (e.g. cosine similarity in word vector space) between the context around the entity mention and the referent entity candidate’s Wikipedia page. The referent entity with larger context similarity score is more likely to be the genuine disambiguation result. The second feature is *semantic relatedness*. It defines correlation measures between a mention’s candidate entity and the unambiguous entities within the same context. The referent entity with a larger correlation score should get more preference.

We notice that both features used by previous research heavily rely on the cross-document hyperlinks in Wikipedia. Semantic relatedness is usually measured in terms of the common incoming hyperlinks to the entities’ Wikipedia pages. For example, as shown in Figure 2(a), the referent entity *Michael I. Jordan* has a high semantic relatedness score with the entity *Andrew Ng*, since they share a lot of common incoming hyperlinks. Meanwhile the context similarity feature implicitly makes use of the hyperlinks to expand entity candidates’ descriptions and then compare them against the mention’s context. For instance, as shown in Figure 2(b), the referent entity *Michael I. Jordan*’s description is expanded by the surrounding context words of its anchor texts. Therefore if we want to perform NED with respect to linkless closed domain knowledge bases, none of these existing algorithms can work well.

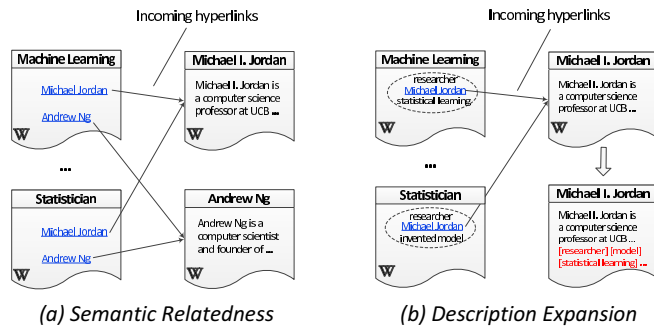


Figure 2: Effects of hyperlinks in NED features

In this paper, we aim at solving the *Named Entity Disambiguation with Linkless Knowledge Bases* (LNED) problem.

We develop a method to automatically mine helpful disambiguation evidences from the reference knowledge base which contains no cross-document hyperlinks. The mined evidences can mimic the role of those links and boost the LNED performance. Mining evidences is not trivial, since without hyperlinks the only labeled data available are the entity candidates’ own description pages. Mentions in other documents are not disambiguated; yet it is still possible to extract new evidences from them, through our model.

Our main contribution is the development of an innovative generative model for mining evidences to mimic the role of cross-document hyperlinks. The harvested evidences can help boost the LNED performance for linkless reference knowledge bases, which are frequently seen in closed domains. To the best of our knowledge, our work is among the first studies on named entity disambiguation with respect to linkless reference knowledge bases. Experimental results show that our proposed method can mine evidences to improve linkless knowledge base’s disambiguation ability and substantially improve the disambiguation accuracy over the baseline approaches.

2. PROBLEM STATEMENT

We formalize the *Named Entity Disambiguation with Linkless Knowledge Bases* (LNED) problem as follows.

Definition 1 (Named Entity Disambiguation with Linkless Knowledge Bases). *Named Entity Disambiguation with Linkless Knowledge Bases (LNED) is the process of associating an entity name mentioned in a text to an entry, representing that entity, in a “linkless” reference knowledge base K . K is comprised of a set of isolated documents D with each document $d \in D$ describing one entity e . There are no¹ cross-document or intra-document hyperlinks among the documents in D .*

In the Big Data and Big Knowledge age, more and more closed domain knowledge bases will emerge and most of them are likely to be linkless. Meanwhile, many domain-specific entity mentions can only be resolved to these knowledge bases. Therefore it is necessary and critical to study the LNED problem and find a good solution to it. In the next section we will describe our approach to tackle the LNED problem via evidence mining.

3. THE EVIDENCE MINING APPROACH

In order to solve the LNED problem, we have to figure out a way to bridge the information gap caused by the absence of cross-document hyperlinks. One straightforward solution is employing an existing NED algorithm (with link-based features removed) to recover the links. Namely, one can perform NED on mentions found in the reference knowledge base and use the disambiguation results to serve as the links. Such a method has a critical drawback. Without hyperlinks, none of the existing NED algorithms can achieve satisfactory results (see Section 6.3 and 6.5). Therefore, a large amount of the recovered links are likely to be incorrect and the features built on these “false links” will do harm to the ultimate disambiguation performance.

¹In this work, we study a general setting with minimal requirements on the underlying knowledge bases. Our proposed method is also applicable to the knowledge bases with a few hyperlinks.

In this work, we propose to bridge the gap by collecting *word-level* disambiguation evidences scattered in the knowledge bases. Compared with the above link-recovery approach, mining fine-grained word evidences has the advantage of being more robust. We will jointly model mention’s link destination (i.e. referent entity) and entity’s supporting evidences (i.e. words) in a probabilistic manner. Instead of explicitly predicting the link, we aim at harvesting some useful word evidences from the mention context, through analyzing the word co-occurrence patterns.

3.1 Documents

The reference knowledge base K is comprised of a set of isolated documents (called *entity documents*), with each document describing one specific entity. For a given target mention m , all the possible entities it can refer to form a candidate entity set. Their main description documents are called *candidate documents*. Since the mention m may also appear in other documents, these additional documents are named as m ’s *mention documents*. Figure 3 illustrates the candidate documents and mention documents of “Michael Jordan”. We aim at mining evidences jointly from these two kinds of documents, to disambiguate m ’s appearances in *query documents*. Below are the brief summaries of these three types of documents.

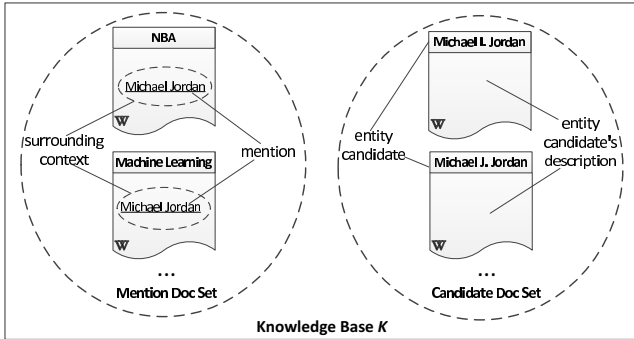


Figure 3: Illustration of mention *Michael Jordan*’s candidate documents and mention documents

1. *candidate documents*: m ’s referent entities’ description documents in the knowledge base. Each document is associated with the corresponding entity it describes.
2. *mention documents*: Other documents in K whose contents contain mention m . These documents could be entity documents with titles different from m .
3. *query documents*: documents containing the target mention m and its query context.

3.2 Word Evidences

The NED problem arises from the fact that the same textual mention can represent multiple different entities depending on the context of its appearance. The reason why context can help disambiguate mention is that each referent entity candidate can be distinguished by a set of representative words. Those representative words can be seen as the disambiguation evidences for those entity candidates. The candidate documents can explicitly provide some basic evidences (i.e. entity descriptions). However, to achieve

good NED performance, we still need some auxiliary information (e.g. semantic relatedness). Therefore, in the LNER problem, we hope to mine additional word evidences from mention documents, to mimic the following effects of the cross-document hyperlinks and thus supply the auxiliary information.

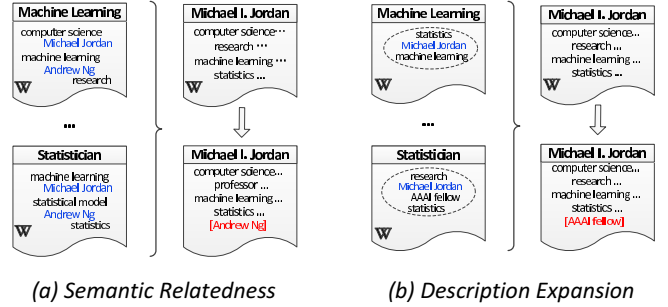


Figure 4: Mimicking the effects of hyperlinks

1. *semantic relatedness*. If two entities are semantically related, they share many common incoming hyperlinks, which can be used to measure their relatedness. Without hyperlinks, we can still capture their relatedness, via adding their names into each other’s supporting word evidences. Then the semantic relatedness effect can be revealed through context comparison. For instance, as shown in Figure 4(a), entity *Michael I. Jordan* and *Andrew Ng* are semantically related, so they co-occur in many documents. Meanwhile, some words (e.g. “research”, “machine learning”) appearing in *Michael I. Jordan*’s descriptions may also appear in these documents. As we know these words are supporting evidences for *Michael I. Jordan*, by analyzing the word co-occurrence patterns, we can associate the words “Andrew Ng” as *Michael I. Jordan*’s disambiguation evidences as well, since they co-occur with *Michael I. Jordan*’s representative words. Now, given a query containing a mention of “Michael Jordan”, with “Andrew Ng” being part of the query context, even we know nothing about the hyperlinks, it is still possible to correctly disambiguate the mention to *Michael I. Jordan*, by comparing the query context with *Michael I. Jordan*’s word-level disambiguation evidences.
2. *description expansion for context similarity*. If entity e_1 appears in the entity document of e_2 , d_{e_2} , via hyperlinks, one can expand e_1 ’s entity document by adding e_1 ’s surrounding words in d_{e_2} . Without hyperlinks, we can still perform such expansions, via directly mining those surrounding words from knowledge base and adding them as e_1 ’s supporting evidences. For instance, as shown in Figure 4(b), the critical descriptive words “AAAI fellow” of entity *Michael I. Jordan* are expanded from a document where *Michael I. Jordan* appears via hyperlink. Now without links, we don’t know to which entity the mention “Michael Jordan” really refers in the document. However, we notice that some words (e.g. “research”, “statistics”) appearing in *Michael I. Jordan*’s descriptions also appear in this document. As we know these words are supporting evidences for *Michael I. Jordan*, by analyzing the word

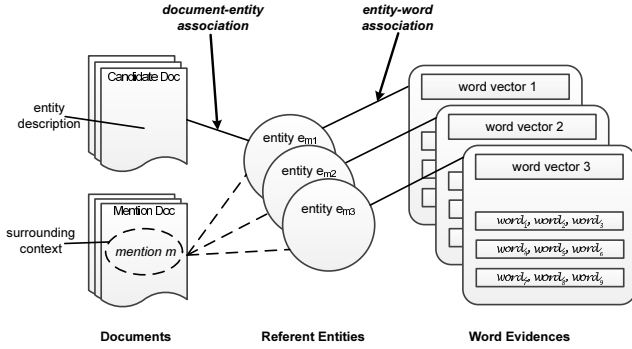


Figure 5: Documents, Entities and Words

co-occurrence patterns, we can associate “AAAI fellow” as *Michael I. Jordan*’s disambiguation evidences. Now, a query containing “AAAI fellow” can be easily disambiguated via context comparison.

Figure 5 shows the association among referent entities, word evidences, and documents. Given a target mention m , the document-entity association θ_d for document d is a distribution over m ’s entity candidates, with each component $\theta_d^{e_{mi}}$ indicating the likelihood that mention m ’s referent entity in d is e_{mi} . Similarly, we have an entity-word association ϕ_e for each entity candidate e so that the words with high probabilities in ϕ_e are precisely the critical disambiguation evidences for e . In Section 4.2 we develop a generative model to automatically learn θ and ϕ .

3.3 LNE D via Evidence Mining

Algorithm 1 provides a high-level description of our approach to tackle the LNE D problem by leveraging the useful disambiguation evidences scattered across the reference knowledge base. We will first generate the entity candidates list for the target mention m . Then we will mine disambiguation evidences jointly from m ’s candidate documents and mention documents, via utilizing the word co-occurrence patterns. Upon the completion of evidence mining, we can utilize the mined evidences to rank entity candidates and choose the top-ranked candidate as disambiguation result.

Note that the evidence mining step (Step 4 in Algorithm 1) is independent of the query context. For each named entity mention m , evidence mining is performed only once, regardless of different query contexts for the same mention. In practice the set of ambiguous mentions can be pre-fetched from the knowledge base K . Therefore the evidence mining step shall run **offline** as a preprocessing step.

4. MINING EVIDENCES

In this section we formally introduce our proposed model, for mining evidences from the knowledge base.

4.1 Model Intuitions

Based on the assumption that disambiguation evidences are entity-specific representative words, it is natural to model

Algorithm 1 LNE D via Evidence Mining

Input: Reference knowledge base K (with no links), named entity mention m , query q .

- 1: Generate candidates list C for mention m
- 2: Fetch candidate documents set D_C from K
- 3: Fetch m ’s mention documents set D_M from K
- 4: Mine evidences from $D_C \cup D_M$
- 5: Use mined evidences to rank candidate $c \in C$ for m in q
- 6: Return top-ranked candidate c_{top} as the answer

each entity as a topic/label and imagine those representative words are generated from such topics. For a given target mention m , we model each of its entity candidates as a regular topic and introduce the following three special topics to capture some noisy or useless words.

1. *background*. Some words in the documents might be general to more than one candidate. Therefore we introduce a special background topic to capture those non-representative words.
2. *undefined*. Since a knowledge base K is very likely incomplete, some entities named after m may not be indexed by K . Therefore we introduce a special topic called “undefined” to capture the words that are associated with these undefined entities.
3. *master*. Since mention documents themselves could be description documents for other entities, words in these documents might be generated from these entities instead of target mention’s candidate entities. Thus we introduce a special “master” topic to capture those words. Note that each mention document will have one unique “master” topic.

4.2 Model Details

We now explain the details of our generative model. Figure 6 shows the graphical structure of dependencies of our model. Each node in the figure corresponds to a random variable or prior parameter. The shaded nodes represent observed variables while other nodes represent latent variables. A plate means the nodes within it are replicated for multiple times. A directed edge from node a to node b indicates that the variable represented by b is dependent on the the variable represented by a .

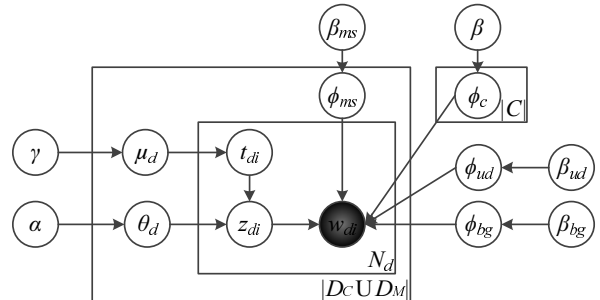


Figure 6: Our Model

Table 1 summarizes the notations used in our model. Given a named entity mention m , we will first find all of its possible referent entity candidates and denote the candidates set

Symbols	Descriptions
D_C	the set of referent entity candidate documents
D_M	the set of mention documents
$D_{surround}$	the set of surrounding context documents (extracted from D_M)
D_{other}	the set of non-surrounding context documents (extracted from D_M)
C	the set of regular entity labels (i.e. entity candidates)
V	vocabulary size
W	surrounding window size of entity mentions
N_d	the number of words in document d
w_{di}	the i -th word of document d
z_{di}	the label associated with w_{di}
t_{di}	the background/regular/master topic indicator for w_{di}
μ_d	the background/regular/master topic proportion for document d
θ_d	the topic distribution for document d
ϕ_{bg}	the word distribution for the background topic
ϕ_{ud}	the word distribution for the undefined topic
ϕ_{ms}	the word distribution for the master topic
ϕ_c	the word distribution for the c -th regular topic ($1 \leq c \leq C $)
α_{ud}, α	the hyperparameters for Dirichlet prior of θ
β_{bg}	the hyperparameter for Dirichlet prior of ϕ_{bg}
β_{ud}	the hyperparameter for Dirichlet prior of ϕ_{ud}
β_{ms}	the hyperparameter for Dirichlet prior of ϕ_{ms}
β	the hyperparameters for Dirichlet prior of ϕ_c ($1 \leq c \leq C $)
γ	the hyperparameters for Dirichlet prior of μ

Table 1: Notations used in our model

as C . Each referent entity candidate will then be treated as a regular topic/label and the total number of them is $|C|$.

- Given a candidate document $d_{candidate}$, its underlying entity e for m is already identified. For each word w in $d_{candidate}$, its label z is either e , or the “background” label. The selection is controlled by an indicator variable t drawn from a multinomial distribution μ .
- For a mention document $d_{mention}$, we further split it into two sub-documents: surrounding context $d_{surround}$ and the rest d_{other} . $d_{surround}$ represents the limited size context (e.g. a width- W word window surrounding m), while d_{other} captures the out-of-window words in $d_{mention}$. For each word w in $d_{surround}$, its label z is chosen from “background”, the “master” entity of $d_{mention}$, or a label (from $C \cup \text{“undefined”}$) drawn from the multinomial distribution θ . The selection is also controlled by an indicator variable t drawn from a multinomial distribution μ . For each word w in d_{other} , its label z is fixed as the “master” entity of $d_{mention}$.

The multinomial distribution θ is drawn from a Dirichlet prior with α and α_{ud} as the hyperparameters. The difference between α and α_{ud} should reflect how conservatively we choose between regular topics and the special “undefined” one. The multinomial distribution μ is drawn from a Dirichlet prior with γ_1, γ_2 and γ_3 as the hyperparameters.

The difference between γ_1, γ_2 and γ_3 should reflect the proportion of “background” topic, regular topics and “master” topics.

For each topic/label in $C \cup \text{“undefined”} \cup \text{“background”} \cup \text{“master”}$, it is associated with a multinomial distribution ϕ over words, which is drawn from the Dirichlet prior with $\beta, \beta_{ud}, \beta_{bg}$ and β_{ms} as the hyperparameters. The difference among $\beta, \beta_{ud}, \beta_{bg}$ and β_{ms} should reflect the content difference among regular labels, the “undefined” label, the “background” label and the “master” labels.

Finally, each word w is drawn from the multinomial distribution ϕ_z , where z is the word label for w . **Our goal** is to infer the document-label association θ and the label-word association ϕ from this model. To summarize, the detailed generative process of our model is as follows:

- Draw the multinomial distribution over words $\phi_c \sim \text{Dirichlet}(\beta)$ for each regular topic c .
- Draw the multinomial distribution over words $\phi_{bg} \sim \text{Dirichlet}(\beta_{bg})$ for the background topic.
- Draw the multinomial distribution over words $\phi_{ud} \sim \text{Dirichlet}(\beta_{ud})$ for the undefined topic.
- Draw the multinomial distribution over words $\phi_{ms} \sim \text{Dirichlet}(\beta_{ms})$ for each master topic.
- For each document $d \in D_C$:
 - Let $e_{candi} =$ candidate label of d
 - Choose a background topic proportion $\mu_d \sim \text{Dirichlet}(\gamma_1, \gamma_2)$.
 - For each word position i in document d :
 - Choose a background indicator $t_{di} \sim \text{Multinomial}(\mu_d)$.
 - if $t_{di} = 0$:
 - Choose topic $z_{di} = bg$.
 - else:
 - Choose topic $z_{di} = e_{candi}$.
 - Choose a word $w_{di} \sim \text{Multinomial}(\phi_{z_{di}})$.
- For each document $d \in D_{surround}$:
 - Draw a topic distribution $\theta_d \sim \text{Dirichlet}(\alpha)$, where $\alpha = (\alpha_{ud}, \alpha_1, \dots, \alpha_c)$ and $\alpha_1 = \dots = \alpha_c = \alpha$.
 - Let $e_{ms} =$ master entity label of the mention document from which d is extracted
 - Choose a background/regular/master topic proportion $\mu_d \sim \text{Dirichlet}(\gamma_1, \gamma_2, \gamma_3)$.
 - For each word position i in document d :
 - Choose a background/regular/master indicator $t_{di} \sim \text{Multinomial}(\mu_d)$.
 - if $t_{di} = 0$:
 - Choose topic $z_{di} = bg$.
 - else if $t_{di} = 1$:
 - Choose topic $z_{di} \sim \text{Multinomial}(\theta_d)$.
 - else if $t_{di} = 2$:
 - Choose topic $z_{di} = e_{ms}$.
 - Choose a word $w_{di} \sim \text{Multinomial}(\phi_{z_{di}})$.
- For each document $d \in D_{other}$:
 - Let $e_{ms} =$ master entity label of the mention document from which d is extracted
 - For each word position i in document d :
 - Choose topic $z_{di} = e_{ms}$.
 - Choose a word $w_{di} \sim \text{Multinomial}(\phi_{z_{di}})$.

4.3 Inference Algorithm

4.3.1 Likelihood Function

The joint likelihood function of our model is:

$$p(\mathbf{w}, \mathbf{t}, \mathbf{z} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \quad (1)$$

$$= \int_{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\mu}} p(\boldsymbol{\theta} | \boldsymbol{\alpha}) p(\boldsymbol{\phi} | \boldsymbol{\beta}) p(\boldsymbol{\mu} | \boldsymbol{\gamma}) p(\mathbf{t} | \boldsymbol{\mu}) p(\mathbf{z} | \boldsymbol{\theta}, \mathbf{t}) p(\mathbf{w} | \mathbf{z}, \boldsymbol{\phi}) d\boldsymbol{\theta} d\boldsymbol{\phi} d\boldsymbol{\mu}$$

Given the hyperparameters $\Gamma = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$, and the observed words \mathbf{w} , we will calculate the posterior probability of $p(\mathbf{t}, \mathbf{z} | \mathbf{w}, \Gamma)$, and use the maximal marginal probability to infer each word’s topic assignment z_{di} and label category indicator t_{di} . After that we can make use of the inferred \mathbf{t} and \mathbf{z} to estimate the document-label association $\boldsymbol{\theta}$ and the label-word association $\boldsymbol{\phi}$.

4.3.2 Approximate Inference via Gibbs Sampling

Similar to many other topic models with conjugate prior (e.g. LDA [2]), exact inference is intractable for our model. Here we use Gibbs Sampling as an approximate inference method. Compared with other approximate inference methods such as Variational Inference, Gibbs Sampling is easy to extend and has been proved to be quite effective in avoiding local optima. In Gibbs Sampling, each hidden variable will be iteratively sampled and the corresponding marginal probability can later be estimated with the samples.

In our model, the word topic assignment variable z_{di} and the label category indicator variable t_{di} are highly correlated since t_{di} controls the selection of z_{di} . Once t_{di} is assigned some value, z_{di} can only be sampled from the corresponding distribution indicated by t_{di} . Therefore we design a blocked Gibbs Sampler to group z_{di} and t_{di} together, and sample from their joint distribution conditional on all other variables, instead of sampling from each one individually.

Algorithm 2 Blocked Gibbs Sampling

```

for all iter from 1 to MaxIter do
  for all  $d \in D_C \cup D_{surround}$  do
    for all  $i$  from 1 to  $N_d$  do
      sample  $\{z_{di}, t_{di}\}$  together according to
       $p(z_{di}, t_{di} | \mathbf{w}, \mathbf{z}_{-di}, \mathbf{t}_{-di}, \Gamma)$ 
    end for
  end for
end for

```

Algorithm 2 describes the blocked Gibbs Sampling process. Note that we will only sample the words in document $d \in D_C \cup D_{surround}$. For words in document $d \in D_{other}$, the word topic is fixed as e_{ms} and therefore no sampling is needed.

The sampling function $p(z_{di}, t_{di} | \mathbf{w}, \mathbf{z}_{-di}, \mathbf{t}_{-di}, \Gamma)$ for different document sets D_C and $D_{surround}$ are only slightly different from each other. Due to the space limit, here we only describe the detailed sampling functions for documents in $D_{surround}$, which is more complicated than those in D_C . For each word w_{di} in $d \in D_{surround}$:

1. z_{di} is sampled to the “background” topic with:

$$p(t_{di} = 0, z_{di} = bg | \mathbf{w}, \mathbf{z}_{-di}, \mathbf{t}_{-di}, \Gamma) \propto \frac{|w = w_{di}, t_w = 0| + \beta_{bg}}{|t_w = 0| + V \cdot \beta_{bg}} \cdot (|t_w = 0, w \in d| + \gamma_1) \quad (2)$$

2. z_{di} is sampled to the “undefined” topic or one of the regular topics with:

$$p(t_{di} = 1, z_{di} = c | \mathbf{w}, \mathbf{z}_{-di}, \mathbf{t}_{-di}, \Gamma) \propto \frac{|w \in d, t_w = 1, z_w = c| + \alpha_c}{|w \in d, t_w = 1| + |C| \cdot \alpha + \alpha_{ud}} \cdot \frac{|w = w_{di}, t_w = 1, z_w = c| + \beta_c}{|t_w = 1, z_w = c| + V \cdot \beta_c} \cdot (|t_w = 1, w \in d| + \gamma_2), \quad (3)$$

where $c \in C \cup \text{“undefined”}$. If $c = \text{“undefined”}$, $\alpha_c = \alpha_{ud}$ and $\beta_c = \beta_{ud}$; otherwise, $\alpha_c = \alpha$ and $\beta_c = \beta$.

3. z_{di} is sampled to the “master” topic with:

$$p(t_{di} = 2, z_{di} = e_{ms} | \mathbf{w}, \mathbf{z}_{-di}, \mathbf{t}_{-di}, \Gamma) \propto \frac{|w = w_{di}, t_w = 2, w \in d| + \beta_{ms}}{|t_w = 2, w \in d| + V \cdot \beta_{ms}} \cdot (|t_w = 2, w \in d| + \gamma_3) \quad (4)$$

Sampling functions for documents in D_C can be derived similarly, according to the corresponding generative process.

4.3.3 Estimating Document-Label Association

After enough iterations of sampling for \mathbf{z} and \mathbf{t} , the document-label association can be estimated by *maximum a posteriori* (MAP) inference:

$$\theta_d^{(c)} = \frac{|w \in d, t_w = 1, z_w = c| + \alpha_c}{|w \in d, t_w = 1| + |C| \cdot \alpha + \alpha_{ud}}, \quad (5)$$

where $c \in C \cup \text{“undefined”}$. If $c = \text{“undefined”}$, $\alpha_c = \alpha_{ud}$ and $\beta_c = \beta_{ud}$; otherwise, $\alpha_c = \alpha$ and $\beta_c = \beta$.

4.3.4 Estimating Label-Word Association

Similarly, we can infer the label-word association by MAP inference:

$$\phi_{bg}^{(v)} = \frac{|w = v, t_w = 0| + \beta_{bg}}{|t_w = 0| + V \cdot \beta_{bg}}, \quad (6)$$

$$\phi_{ud}^{(v)} = \frac{|w = v, t_w = 1, z_w = ud| + \beta_{ud}}{|t_w = 1, z_w = ud| + V \cdot \beta_{ud}}, \quad (7)$$

$$\phi_c^{(v)} = \frac{|w = v, t_w = 1, z_w = c| + \beta}{|t_w = 1, z_w = c| + V \cdot \beta} \quad (8)$$

For the LNEED task, we are particularly interested in the label-word association $\phi_c^{(v)}$, which reveals the disambiguation evidences for each referent entity candidate.

5. RANKING REFERENT CANDIDATES

Our ultimate goal for the LNEED problem is to disambiguate entity mentions in query documents. Upon the completion of the evidence mining step, we can make use of the knowledge learned from our evidence mining model to rank referent entity candidates, and choose the top-ranked candidate as disambiguation result. Given a query document, we predict its word labels \mathbf{z} using the incremental Gibbs Sampling algorithm described in [17]. Namely, we iteratively update the word topic assignments of a query document using the above inference process, but with the previously learned global knowledge (i.e. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$) fixed. As the sampling is operated only on the words in the query document, it converges very fast (e.g. less than 30 iterations).

After the sampling converges, we infer the document-label association θ_d for each query document d , using Equation 5. The disambiguation result can then be predicted with the maximal marginal probability:

$$LNEED(d) = \operatorname{argmax}_c \theta_d^{(c)}. \quad (9)$$

6. EXPERIMENTS

In this section, we evaluate the effectiveness of our proposed method for the L_NED problem on two real-life query datasets [18]: one from news, and the other from Twitter. We will: (1) illustrate the effectiveness of mining evidences for L_NED, by comparing our model against a similar generative model which has no evidence mining component; (2) demonstrate the superiority of our method by comparing it with a baseline method which also performs NED via evidence mining; (3) compare the end-to-end disambiguation accuracy of our method, with two state-of-the-art NED methods (with their link-based features disabled); (4) show how the performance of our method changes with respect to surrounding window size of entity mentions. All the experiments, if not specifically mentioned, are conducted on a server with 2.40GHz Intel Xeon CPU and 48GB RAM.

6.1 Datasets

We adopt the two datasets used in [18] to test our method. The first one is derived from the TAC-KBP2009 dataset, which is created for the Entity Linking task in the Knowledge Base Population track at the Text Analysis Conference. The queries in this dataset are all news articles. Therefore the queries are relatively long and the writing quality is good. *Note that our experiments setting is more challenging than the TAC-KBP competition [28] since we don't assume the availability of various kinds of annotations (e.g. entity type, Wikipedia infobox).* The second dataset is generated from Twitter. Since tweets have the 140-character constraint and the words used in them are often irregular, the queries are usually very short and the writing quality is not well-expected. Table 2 shows some basic statistics of these two datasets. As can be seen, it is quite challenging to conduct NED on these two datasets since there are many entity candidates for the query mentions.

	TAC-KBP2009	Twitter
# of Queries	424	340
Avg Length of Queries	53.15 words	16.46 words
Avg # of Candidates	24.024	19.279

Table 2: Basic statistics of test datasets

6.2 Experiments Setup

While more and more closed domain knowledge bases are emerging, most of them are restricted to inside domain access and thus not publicly available. Moreover, to make it practical to compare with previous methods which use Wikipedia as the knowledge base, the *linkless version of Wikipedia* is used in this work, where we keep all entity description pages in Wikipedia but discard all link-related information such as cross-document hyperlinks and entity categories. In such a case, for each referent entity candidate, the only labeled data we have is its own Wikipedia page. This modification can establish a **public benchmark** for algorithm comparison. Following previous work [26, 23, 4] on NED, we merely make use of the “Disambiguation Pages” and “Redirect Pages” in Wikipedia to find all the entity candidates that a given mention can be mapped to. Note that these two types of pages are not related with the cross-document hyperlinks. For fetching mention documents from the knowledge base, we make use of the Wikipedia

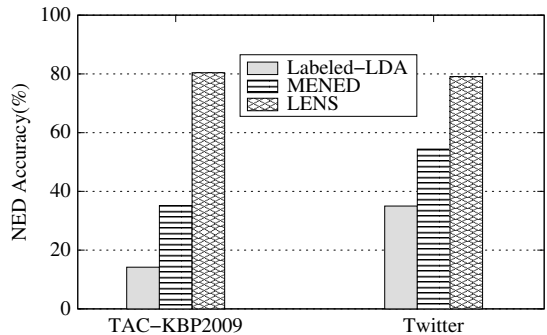


Figure 7: LENS vs. Labeled-LDA and MENED

Search API and collect all the Wikipedia pages that contain the exact query mention.

Our model has some hyperparameters, α , β and γ . In this work, we use the following parameter settings: $\alpha = 0.01$, $\alpha_{ud} = 0.1$, $\beta = 0.01$, $\beta_{ud} = 0.1$, $\beta_{bg} = 0.1$, $\beta_{ms} = 0.01$, $\gamma_1 = 0.01$, $\gamma_2 = 1$, $\gamma_3 = 2$. These parameters are tuned on a separate develop dataset containing 15 queries and then reused in all the experiments without any further tuning. Besides, the surrounding window size W (see Section 4.2) is set as 40 for Twitter dataset and 30 for TAC-KBP2009 dataset.

To train the evidence mining model, we run 2000 iterations of our Gibbs sampling algorithm to its convergence. The training time varies from several seconds to a few hours for different entity mentions, depending on the corresponding number of candidate documents and mention documents. While we do aware that the model training is inefficient for some highly ambiguous mentions, our main focus for this work is the disambiguation accuracy and therefore we leave the efficiency and scalability aspects for future study. As discussed in Section 3.3, this training process shall run offline in practice. Therefore the training time shall not cause critical problem to the real world applications. After training, the online disambiguation of query documents (Section 5) is very quick (usually within seconds).

6.3 Effectiveness of Evidence Mining

We first illustrate the effectiveness of mining evidences to bridge the information gap (caused by the missing links), by comparing the L_NED accuracy of our model (we name it as *Linking Evidences in Not Well Linked Sources, LENS*), with a baseline model, **Labeled-LDA** [22], which has no evidence mining component. Labeled-LDA is also extended from the standard LDA [2]. In Labeled-LDA, each document can have multiple labels and the label-word correspondences can be inferred. Labeled-LDA can be directly used for L_NED purpose by treating each referent entity candidate as a unique label. In this way, the model can learn the label-word association from the candidate documents and then use it to rank referent entities with respect to the query. Note that Labeled-LDA’s disambiguation decisions are made from candidate documents. All other documents containing mentions are discarded without processing.

Figure 7 shows the results for LENS and Labeled-LDA on both datasets. Suffering from the information gap caused by the missing links, Labeled-LDA works badly in L_NED task. Conversely, our LENS model works much better. That is because LENS utilizes not only the candidate documents, but

also useful evidences scattered across mention documents. The performance gain clearly illustrates the effectiveness of mining evidences to bridge the information gap.

6.4 Comparison of Evidence Mining Methods

We then conduct experiments to compare the disambiguation accuracy of LENS, against the **MENED** model proposed in [18]. **MENED** was originally designed to mine additional evidences from external corpus to help NED. In the **LNED** problem setting, we can treat only the candidate documents as “internal corpus” and all the mention documents as the “external corpus”. By doing so, **MENED** can perform **LNED** via mining evidences from those mention documents. *Note that **MENED**’s experiment setting here is quite different from the original one discussed in [18]. In [18], both candidate documents and mention documents are considered as labeled documents since the entity mentions in mention documents are naturally labeled by the hyperlinks. In contrast, in the **LNED** setting, mention documents become unlabeled due to the missing of hyperlinks.*

Figure 7 shows that LENS outperforms **MENED** on both datasets. The reasons are two fold: (1) In **MENED**, each document has only one label. The model will first sample the document label. Once the document label is chosen, every word in the document can only have two possible labels: foreground or background, and the foreground label is restricted by the document label. While this is an effective constraint for the problem studied in [18], it is inappropriate for the **LNED** problem. In **LNED**, the number of labeled documents is much less than that of unlabeled documents. Therefore it is very likely to assign a wrong label to an unlabeled document. If this constraint is applied, then all the words inside that document will get wrong labels, which will in turn confuse the label-word association and get more documents wrongly labeled. To avoid this, LENS models each document as a mixture of different labels and directly infer the label for each word. (2) In **MENED**, every word within a mention document is considered as either a background word, or a supporting evidence for one of the entity candidates. In **LNED** problem, each mention document itself is a description page for some “master” entity. Hence, some words in the mention document may be generated from the “master” entity instead of one of the entity candidates. To properly handle these words, in LENS we introduce a special “master” topic and assume all such words are generated from the corresponding “master” topics.

6.5 End-to-end NED Accuracy

We then conduct experiments to compare the end-to-end disambiguation accuracy of LENS, against two state-of-the-art NED methods: **Wikifier** [23], a widely-used NED system using a machine learning based hybrid strategy to combine various kinds of features together, and **AIDA** [15], a robust NED system making use of weighted mention-entity graph to find the best joint mention-entity mapping. To make Wikifier and AIDA fit the **LNED** problem setting, we modified them to disable all the link-based features (e.g. semantic relatedness) and then retrained the models. The modified linkless versions are denoted as **Wikifier(w/o link)** and **AIDA(w/o link)**². All three methods use a Wikipedia repository of late 2012 as the reference knowledge base.

²AIDA extracts keyphrases from various sources to describe entities. As these keyphrases are pre-extracted and indexed

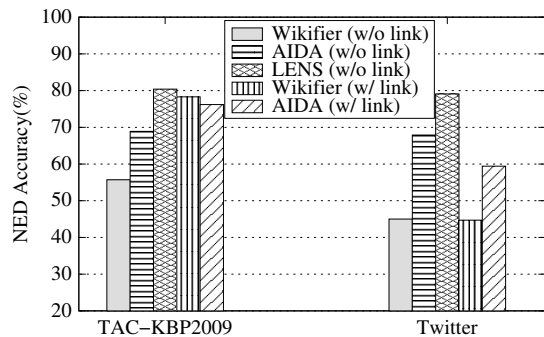


Figure 8: LENS vs. Wikifier vs. AIDA

Figure 8 shows that LENS significantly outperforms Wikifier and AIDA on both datasets. Compared with Wikifier and AIDA, LENS can collect disambiguation evidences scattered in the reference knowledge bases to mimic the role of the missing cross-document hyperlinks and thus achieve better disambiguation results. To demonstrate the helpfulness of such links, we also present the NED accuracy of the original Wikifier and AIDA (with all features enabled and using the parameter settings suggested by their authors) in Figure 8 (denoted as **Wikifier(w/ link)** and **AIDA(w/ link)**, respectively). As we can see, on the TAC-KBP2009 dataset, the link-based features are quite helpful in boosting the NED accuracy. LENS can harvest evidences to mimic the effects of these links and achieve similar NED accuracy with the original Wikifier and AIDA. On the Twitter dataset, the full version Wikifier and AIDA perform even worse than the modified linkless versions. This is because both Wikifier and AIDA utilize entity popularity information to give more preferences to popular entity candidates, while in Twitter dataset, many queries’ disambiguation answers are actually non-famous entities. LENS can still achieve good performance on the Twitter dataset, due to the following two reasons: (1) LENS makes NED decisions without applying any prior preferences; (2) the evidences mined from “description expansion” are very helpful for short texts like tweets.

6.6 Impact of Surrounding Window Size

In this part we conduct experiments to illustrate how the performance of our LENS model changes with respect to the surrounding window size W in mention documents (see Section 4.2). Here we still use the same parameter settings as in previous experiments. Figure 9 shows that as W increases, the NED accuracy will increase to a peak point and decrease afterwards. There are two factors inside this phenomenon. As the window size increases, more evidences are exposed to the model. Hence, the NED accuracy will increase. On the other hand, with the increase of W , more noisy words may be wrongly judged as supporting evidences. Our model incorporates the “master” topic to filter the help-less words. However, when the window is too large, most words in the mention document $d_{mention}$ will be split into $d_{surround}$. Therefore the very few words in d_{other} are insufficient to filter out the noisy words in $d_{surround}$, which results in the performance degradation. In practice, W usually works very well at $20 \sim 40$.

offline, we are unable to modify them. So precisely speaking, AIDA(w/o link) still utilizes few link-based features.

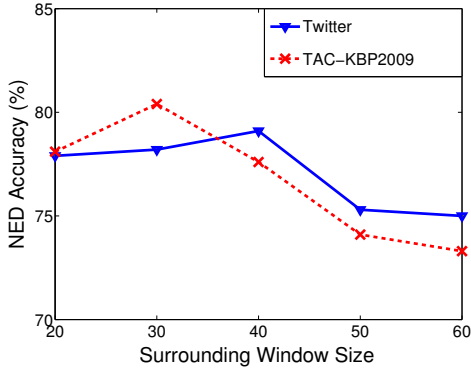


Figure 9: Varying Surrounding Window Size

7. RELATED WORK

Named entity disambiguation (NED) has received a lot of attentions in recent years. Approaches that disambiguate named entity mentions with respect to Wikipedia date back to Bunescu et. al’s work [3]. They defined a similarity measure to compute the cosine similarity between the text around the entity mention and the referent entity candidate’s Wikipedia page. The referent entity with the maximum context similarity score is selected as the disambiguation result. Hoffart et al [14] proposed a method to mine salient phrases for each entity, and then compare the overlap of such phrases for NED. They utilized “internal and external links” to harvest phrases and relies on links among entity pages for measuring salience. Several subsequent work incorporated more information into similarity comparison: Gottipati et. al [10] explored query expansion, while Zhang et. al [29] considered acronym expansion. To incorporate different types of disambiguation knowledge together, Han et. al [11] proposed a generative model to include information from entity popularity, mention-entity association and context similarity in a holistic way. And to overcome the deficiency of the bag of words model, Sen [24] adopted a latent topic model to learn the context-entity association to help disambiguation. In that work, the cross-document hyperlinks in Wikipedia are utilized to provide the labeled training data. Cucerzan’s work [7] is the first one to realize the effectiveness of using semantic relatedness to help named entity disambiguation. In that work, the semantic relatedness between the referent entity candidate and other entities within the same context is calculated based on their overlaps in categories and incoming links in Wikipedia. Milne et. al [19] refined Cucerzan’s work by defining semantic relatedness using Normalized Google Distance [6] and only using “unambiguous entities” in the context to calculate semantic relatedness. Recently, several methods [13, 15, 23, 26, 29] also tried to combine together “context similarity” and “semantic relatedness” using a hybrid strategy which could further improve the NED accuracy.

Almost all these previous NED algorithms use Wikipedia as the reference knowledge base. However, most not well known or domain specific entities are not captured by Wikipedia. To solve this problem, Sil et. al [27] proposed the Open-DB NED problem, which is to resolve an entity to any relational database that meets mild conditions about data format. They investigated a distant supervision approach

and a domain adaptation approach to leverage the structural information in the reference relational databases. Their experiments on the movie and sports domain demonstrated their method’s effectiveness. Similarly, Zheng et. al [30] studied disambiguating entity mentions to Freebase, Jin et. al [16] investigated linking entity mentions to a people profile database and Pantel et.al [21] addressed the task of associating Web search queries with entities from a product catalog. Recently, Shen et. al [25] proposed a probabilistic model to link entity mentions in Web text to DBLP bibliographic network. All these work used some schema-rich databases or networks as the reference knowledge bases, and made use of the structural information to help perform disambiguation. Different from them, our work focuses on a more challenging problem setting where the reference knowledge base is a set of noisy, unstructured and isolated text documents. Instead of directly applying the information provided by the relational databases, our method has to mine useful disambiguation evidences from the knowledge base and use them to bridge the information gap caused by the absence of cross-document links. Our work is also different from Cai et. al [4]’s and Chisholm et. al [5]’s work on link enrichment for named entity disambiguation. In their work, the goal is to add more cross-document links to Wikipedia via using the co-occurrence of the existing links [4] or using Web links [5]. While in our work, none of the existing links are available and we have to mine evidences completely out of the linkless documents.

Mining evidences for NED was studied by Li et. al [18]. In that work, the goal is to mine useful evidences from external corpus to bridge the gap between the keywords in a query and the reference knowledge base. They proposed a semi-supervised approach to extract evidences from unlabeled external documents, via leveraging the labeled information (the entity descriptions plus the context within which entities are hyperlinked to in Wikipedia) in the knowledge base. Our work also tries to mine evidences to bridge the linkage gap. However, it is much more difficult since the majority of the labeled information used in [18] becomes unavailable in our case, due to the lack of the cross-document hyperlinks. The problem studied in [18] is orthogonal to our work. After the link-originated evidences are harvested through our model, the method proposed in [18] can be applied to mine more evidences from external corpus to further enhance the knowledge base’s disambiguation ability.

Our proposed model is inherited from the Latent Dirichlet Allocation (LDA) model. LDA was first proposed by Blei et. al [2] for finding the document-topic association and the topic-word association in text documents. Ramage et. al [22] extended LDA to Labeled-LDA so that each document can have multiple labels and the label-word correspondences can be inferred. Different from both LDA and Labeled-LDA, our model is particularly designed for the LINED task and it has different generative processes for different types of documents.

8. CONCLUSIONS

In this paper, we studied the problem of *Named Entity Disambiguation with Linkless Knowledge Bases* (LINED). We proposed a generative model to automatically mine useful evidences from the reference knowledge base so that the mined evidences can help mimic the role of the missing links. With a specific modeling of “background topic”, “undefined

entities” and “master entities”, our model is able to harvest useful evidences from noisy text. To evaluate the effectiveness of our model, a thorough experimental study was conducted. The experimental results demonstrated that our proposed method can mine useful evidences to bridge the information gap caused by the absence of cross-document links, thus significantly boosting the NED performance for a linkless knowledge base.

9. ACKNOWLEDGMENTS

This research was sponsored in part by the Army Research Laboratory under cooperative agreements W911NF-09-2-0053 and NSF IIS-0954125. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

10. REFERENCES

- [1] <http://en.wikipedia.org/wiki/wikipedia:wikipedians>.
- [2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [3] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, pages 9–16, 2006.
- [4] Z. Cai, K. Zhao, K. Q. Zhu, and H. Wang. Wikification via link co-occurrence. In *Proceedings of CIKM*, pages 1087–1096, 2013.
- [5] A. Chisholm and B. Hachey. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156, 2015.
- [6] R. Cilibrasi and P. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.
- [7] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, pages 708–716, 2007.
- [8] N. Dalvi, R. Kumar, and B. Pang. Object matching in tweets with spatial models. In *Proceedings of WSDM*, pages 43–52, 2012.
- [9] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *Proceedings of ICCL*, pages 277–285, 2010.
- [10] S. Gottipati and J. Jiang. Linking entities to a knowledge base with query expansion. In *Proceedings of EMNLP*, pages 804–813, 2011.
- [11] X. Han and L. Sun. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of ACL-HLT*, pages 945–954, 2011.
- [12] X. Han and L. Sun. An entity-topic model for entity linking. In *Proceedings of EMNLP*, pages 105–115, 2012.
- [13] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of SIGIR*, pages 765–774, 2011.
- [14] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of CIKM*, pages 545–554, 2012.
- [15] J. Hoffart, M. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of EMNLP*, pages 782–792, 2011.
- [16] Y. Jin, E. Kıcıman, K. Wang, and R. Loynd. Entity linking at the tail: Sparse signals, unknown entities and phrase models. In *Proceedings of WSDM*, 2014.
- [17] S. Kataria, K. Kumar, R. Rastogi, P. Sen, and S. Sengamedu. Entity disambiguation with hierarchical topic models. In *Proceedings of SIGKDD*, pages 1037–1045, 2011.
- [18] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan. Mining evidences for named entity disambiguation. In *Proceedings of SIGKDD*, pages 1070–1078, 2013.
- [19] D. Milne and I. Witten. Learning to link with wikipedia. In *Proceedings of CIKM*, pages 509–518, 2008.
- [20] A. A. Morgan, Z. Lu, X. Wang, A. M. Cohen, J. Fluck, P. Ruch, A. Divoli, K. Fundel, R. Leaman, J. Hakenberg, et al. Overview of biocreative ii gene normalization. *Genome biology*, 9(Suppl 2):S3, 2008.
- [21] P. Pantel and A. Fuxman. Jigs and lures: Associating web queries with structured entities. In *Proceedings of ACL-HLT*, pages 83–92, 2011.
- [22] D. Ramage, D. Hall, R. Nallapati, and C. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings EMNLP*, pages 248–256, 2009.
- [23] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of ACL*, pages 1375–1384, 2011.
- [24] P. Sen. Collective context-aware topic models for entity disambiguation. In *Proceedings of WWW*, pages 729–738, 2012.
- [25] W. Shen, J. Han, and J. Wang. A probabilistic model for linking named entities in web text with heterogeneous information networks. In *Proceedings of SIGMOD*, 2014.
- [26] W. Shen, J. Wang, P. Luo, and M. Wang. Linden: linking named entities with knowledge base via semantic knowledge. In *Proceedings of WWW*, pages 449–458, 2012.
- [27] A. Sil, E. Cronin, P. Nie, Y. Yang, A.-M. Popescu, and A. Yates. Linking named entities to any database. In *Proceedings of EMNLP/CoNLL*, pages 116–127, 2012.
- [28] V. Varma, V. Bharat, S. Kovelamudi, P. Bysani, G. Santosh, K. Kumar, K. Reddy, K. Kumar, and N. Maganti. Iiit hyderabad at tac 2009. In *Proceedings of TAC*, 2009.
- [29] W. Zhang, Y. Sim, J. Su, and C. Tan. Entity linking with effective acronym expansion, instance selection and topic modeling. In *Proceedings of IJCAI*, pages 1909–1914, 2011.
- [30] Z. Zheng, X. Si, F. Li, E. Y. Chang, and X. Zhu. Entity disambiguation with freebase. In *Proceedings of WI-IAT*, pages 82–89, 2012.