Mining Evidences for Named Entity Disambiguation

Yang Li[1], Chi Wang[2], Fangqiu Han[1], Jiawei Han[2], Dan Roth[2], Xifeng Yan[1]
{yangli, fhan, xyan}@cs.ucsb.edu, {chiwang1, hanj, danr}@cs.uiuc.edu

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

[2]Department 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 knowledge base such as Wikipedia. Such disambiguation can help enhance readability and add semantics to plain text. It is also a central step in constructing high-quality information network or knowledge graph from unstructured text. Previous research has tackled this problem by making use of various textual and structural features from a knowledge base. Most of the proposed algorithms assume that a knowledge base can provide enough explicit and useful information to help disambiguate a mention to the right entity. However, the existing knowledge bases are rarely complete (likely will never be), thus leading to poor performance on short queries with not well-known contexts. In such cases, we need to collect additional evidences scattered in internal and external corpus to augment the knowledge bases and enhance their disambiguation power. In this work, we propose a generative model and an incremental algorithm to automatically mine useful evidences across documents. With a specific modeling of "background topic" and "unknown entities", our model is able to harvest useful evidences out of noisy information. Experimental results show that our proposed method outperforms the state-of-the-art approaches significantly: boosting the disambiguation accuracy from 43% (baseline) to 86% on short queries derived from tweets.

Categories and Subject Descriptors

I.7 [**Computing Methodologies**]: Document and Text Processing

Keywords

Entity Disambiguation, Evidence Mining, Generative Model, Knowledge Expansion, Semi-supervised Learning

1. INTRODUCTION

Access to an organized information network or knowledge graph is critical for many real-world tasks. Most real-world information is unstructured, interconnected, noisy, and often expressed in the form of text. This inspires constructing an organized, semi-structured information network from the large volume of noisy text data. Such formal and structural representation of information has the advantage of being easy to manage and reason with, which can greatly facilitate many Artificial Intelligence applications, such as Semantic Search, Reasoning and Question Answering. To achieve this goal, knowledge bases such as DBpedia [1], Freebase [4] were manually constructed. However, due to the laborious, time consuming, and costly extracting and labeling process, these knowledge bases are often restricted by a very limited coverage. Recently, automatically constructed knowledge networks including YAGO [27], NELL [6], Reverb [12], have emerged. But unfortunately, they suffer from the problems of low coverage [27, 6], or poor quality [12]. How to automatically construct a high-quality information network from a large amount of unstructured and noisy text data remains an open research problem.

An important component in constructing information networks is *named entity disambiguation (NED)*. Given the named entity mentions in unstructured text data, the goal of NED is to map them to their corresponding real world entities in a knowledge base such as Wikipedia. Different from *entity resolution (ER)*, whose goal is to cluster entity mentions into several disjoint groups with each group representing a unique entity, NED requires explicitly identifying which underlying entity a given named entity mention should refer to. The NED task is challenging due to the fact that many named entity mentions are ambiguous: the same mention can refer to various different real world entities when they appear in different contexts. For example, "Michael Jordan" can refer to the basketball star in NBA, the Machine Learning researcher in Berkeley or some other people. NED plays a critical role in high-quality information network construction. When new information extracted from text data is ready to be inserted into the network, it is necessary to know which real world entity this piece of information should be associated with. If the system makes a wrong decision here, the network will not only lose some information, but also introduce errors. For example, as shown in Figure 1, if the extracted information *"elected as AAAI fellow"* is wrongly associated with the basketball player *Michael Jordan*, the network will lose the information that *Michael Jordan (Machine Learning)* is an AAAI

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$15.00.

fellow, as well as wrongly including *Michael Jordan (Basketball Player)* as a fellow of AAAI.

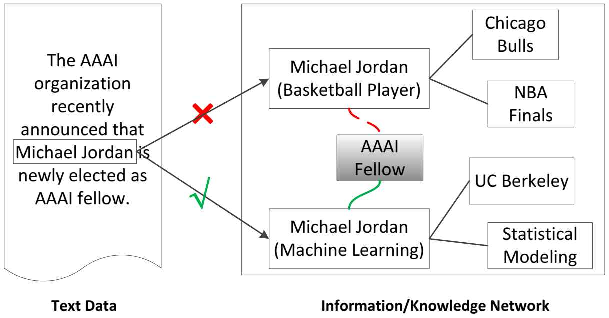
The AAAI organization recently announced that Michael Jordan is newly elected as AAAI fellow.
Michael Jordan (Basketball Player)
Chicago Bulls
NBA Finals
AAAI Fellow
UC Berkeley
Michael Jordan (Machine Learning)
Statistical Modeling
Text Data
Information/Knowledge Network

Figure 1: Named Entity Disambiguation Example

In recent years, the NED task has received a lot of research interests. Many methods [10, 11, 17, 18, 19, 21, 24, 25, 26, 28] have been proposed to disambiguate named entity mentions in free text with respect to Wikipedia. Generally speaking, three kinds of features are explored by those methods. The first one is a statistical feature called *entity popularity*. It is based on the assumption that the most prominent entity for a given entity mention is the most probable underlying entity for that mention. Usually the "most prominent" entity is defined as the entity which uses the mention most frequently as a hyperlink anchor text in Wikipedia. Previous study [24] has shown that this simple heuristic is a very reliable indicator of the correct disambiguation. But obviously, methods merely depending on this feature are not robust as they will disambiguate all appearances of an entity mention to a fixed entity, regardless of the context along with it.

The second feature is a textual feature called *context similarity*. It takes the entity mention's context into consideration and defines similarity measures between the text around the entity mention and the document describing the referent entity in Wikipedia. This feature complements the *entity popularity* prior and is widely used in almost every method. One problem with *context similarity* is that it requires exact word overlap between the two compared texts, which may become an over-strict constraint due to natural language's usage flexibility. To handle this problem, the third feature "topical coherence" is proposed. This feature is a structural feature making use of Wikipedia's cross-page links to define two entities' topical coherence. The intuition for using this feature is that the mention's referent entity should be topical coherent with other entities within the same context. Previous study [24] has proved the effectiveness of using this feature. Recently, several methods [18, 19, 24, 26, 28] also tried to combine together all these three features using a hybrid strategy which can further improve the accuracy.

Almost all of the previously proposed algorithms assume that the knowledge base can provide enough explicit and useful information to help disambiguate a mention to the right entity. However, in many situations, the information contained in the knowledge base is insufficient, thus leading to a connection gap between the keywords in a query (a named entity mention along with its context) and the knowledge base. Note that such situations are not rare since the reference knowledge base (e.g. Wikipedia) has a limited coverage and therefore cannot capture every aspect of a referent entity. In these cases, the existing state-of-the-art methods will fail to make correct disambiguation decisions because there is not enough information for them to utilize. The following two examples show the cases where key evidences (*"Eric Xing"* and *"paper"*, respectively) are not available (*Example1*) in the knowledge base or overwhelmed (*Example2*) by other evidences (*"won, best, award"*).

EXAMPLE 1. *Eric Xing worked with Michael Jordan from 1999 to 2004.*

EXAMPLE 2. *Michael Jordan won the best paper award.*

To solve the above problem, we need to collect additional evidences scattered in internal and external corpus to augment the knowledge base and enhance its disambiguation power. Mining additional evidences is an effective method for improving NED performance because it helps address at least two types of failures in existing approaches:

1. *No evidence failure*, i.e. the knowledge base does not cover the information contained in the query. Evidence mining helps by adding that information into the knowledge base. For instance, as shown in Example 1, the knowledge base contains no information about "Eric Xing," therefore the existing methods have no idea which entity the mention "Michael Jordan" should refer to. With the help of evidence mining, we can directly add "Eric Xing" as a supporting evidence of "Michael Jordan (Machine Learning)", via analyzing a large amount of documents outside the knowledge base, thus making the disambiguation an easy task.

2. *Insufficient evidence failure*, i.e. the important disambiguation evidences appear rarely in the knowledge base. Evidence mining again helps by increasing the weight of those evidences. For instance, as shown in Example 2, the most important disambiguation evidence here would be "paper". But since the occurrence of "paper" in "Michael Jordan (Machine Learning)" is not as frequent as the occurrences of "won", "best" and "award" in "Michael Jordan (Basketball Player)", the existing methods may wrongly disambiguate the mention "Michael Jordan" to the basketball player. With the help of evidence mining, we can give more weight to "paper" and thus avoiding such mistakes.

In this paper, we aim at developing a method to automatically mine helpful evidences from internal and external corpus to boost the NED performance. Mining external evidences is much harder than mining internal ones, since internal documents in a knowledge base are well labeled and linked. Mentions in external documents are not disambiguated; yet it is still possible to extract new evidences from them, through our model. Our method can incrementally enrich the useful evidence set, making use of information both inside and outside the reference knowledge base. With a specific modeling of "background topic" and "unknown entities", our method can harvest helpful evidences out of noisy information.

Our main contribution is the development of an innovative generative model and a novel incremental algorithm for mining additional evidences to help boost the NED performance. To the best of our knowledge, our study is the first work on mining evidences for named entity disambiguation, a critical problem in constructing high-quality information network. Experimental results show that our proposed method can mine additional evidences to significantly improve knowledge base's disambiguation ability. Our work is also useful to the work on developing new NED algorithms and the mined evidences can be beneficial to any such algorithms.

2. PROBLEM STATEMENT

We formalize the *Named Entity Disambiguation (NED)* problem and our *Mining Evidences for Named Entity Disambiguation (MENED)* task as follows.

Definition 1 (Named Entity Disambiguation). *Named Entity Disambiguation (NED) is the process of associating an entity name mentioned in a text to an entry, representing that entity, in a knowledge base (e.g. Wikipedia). Given a textual named entity mention m along with the unstructured text t in which it appears, and a reference knowledge base K, the goal is to produce a mapping from the mention m to its referent real world entity e in K.*

Definition 2 (Mining Evidences for NED). *Mining Evidences for Named Entity Disambiguation (MENED) is the task of finding additional evidences inside and outside the knowledge base to improve NED accuracy. Given a textual named entity mention m, a reference knowledge base K, and a document corpus C outside K, the task is to mine additional evidences from K and C which can further help the disambiguation of m with respect to K.*

The MENED task is independent of the query context. For each named entity mention m, MENED is performed only once, regardless of different query contexts for the same mention. In practice the set of solvable ambiguous mentions can be pre-calculated from the knowledge base K (e.g. the whole set of entities indexed by K). Therefore the MENED process shall run **offline as a preprocessing step**. After MENED, any NED algorithm can make use of the evidences harvested by MENED to disambiguate m. In this work, a component in our MENED model can be reused to perform NED directly (Section 3.4.3).

3. MODEL & ALGORITHM

In this section we formally introduce our proposed model and algorithm, for mining new evidences to help named entity disambiguation. We will first describe the intuitions behind our model, then provide details about how the model is constructed and how the incremental algorithm works, and finally we will discuss how to perform inference on our model to estimate the document-label(entity) association and the label(entity)-word association.

3.1 Intuitions Behind the Model

We first describe the intuitions behind our model, before detailing the model in the next section. The goal of named entity disambiguation is to find a named entity mention's referent entity by utilizing the context along with the mention. The reason why context can help disambiguation 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. Therefore it is natural to model each entity as a topic/label[1] and imagine those representative words are generated from such topics. Since we are only interested in those representative words which are highly related to the underlying entities, we model each entity mention's limited size context as a document. Each document can be associated with only one topic/label corresponding to its entity mention's real referent entity.

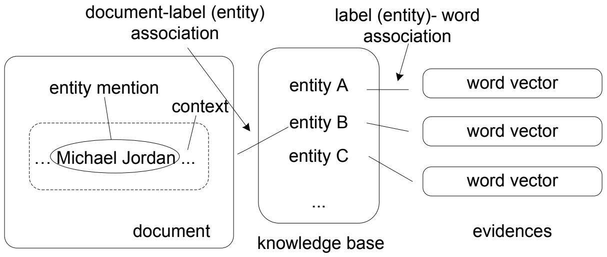

Figure 2: Entity, Word (Evidence), and Document

Though we have adopted the limited size context constraint to ensure topic centrality, some words within the context may still be general to some or all topics/labels. To specifically model this phenomenon, we introduce a special background topic to capture those non-representative words. On the other hand, we also notice that sometimes we may encounter documents whose underlying entities are not within the referent entity candidates. This is due to the fact that currently there is no perfect solution to generate complete referent entity candidates for a given entity mention. Obviously it is not appropriate to assign any topic/label to these documents. Therefore we introduce another special topic called "default" to capture words from the documents with unknown or unsure underlying entities. With all these intuitions, we are able to properly model the document-label association and the label-word association. Figure 2 shows the association among entities, words, and documents. Evidences are reflected by the words and their association strengths with entities, after discounting "background" and "default" topics. In the next section we will introduce our proposed generative model based on these intuitions.

3.2 Model Details

We now explain the details of our generative model. Figure 3 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.

Table 3.2 summarizes the notations used in our model. Given a named entity mention m, we will first generate all of its possible referent entity candidates. Following previous work [26, 24] on NED, we make use of the structural information of Wikipedia to find all the entities that m can be mapped to. Each referent entity candidate will be treated as a regular topic/label and the total number of them is K. We denote the set of regular topics/labels as S. For each occurrence of m, we model its limited size context (e.g. a

[1] In this paper we use "entity", "topic" and "label" interchangeably for describing our model.

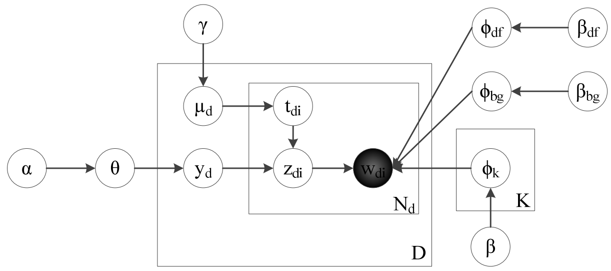

Figure 3: Our Model for MENED

Symbols	Descriptions
D	the set of documents (e.g. named entity mention's limited size context)
K	the number of referent entity candidates
S	the set of regular entity labels
N_d	the number of words in document d
w_{di}	the i-th word of document d
z_{di}	the label associated with the i-th word of document d
y_d	the label associated with document d
t_{di}	the background indicator for the i-th word of document d
μ_d	the background topic proportion for document d
θ	the topic/label distribution
ϕ_{bg}	the word distribution for the background topic/label
ϕ_{df}	the word distribution for the default topic/label
ϕ_k	the word distribution for the k-th regular topic/label ($1 \leq k \leq K$)
α_{df}, α	the hyperparameters for Dirichlet prior of θ
β_{bg}	the hyperparameter for Dirichlet prior of ϕ_{bg}
β_{df}	the hyperparameter for Dirichlet prior of ϕ_{df}
β	the hyperparameters for Dirichlet prior of ϕ_k ($1 \leq k \leq K$)
γ	the hyperparameters for Beta prior of μ

Table 1: Notations used in our model

width-W word window surrounding m) as a document. For a labeled document (e.g. the document in which the genuine underlying entity for m is already identified), its document label y is fixed and within S. For an unlabeled document (e.g. the document in which the genuine underlying entity for m is not clear yet), its document label y is drawn from $S \cup$ "*default*", according to a multinomial distribution θ, which itself is drawn from a Dirichlet prior with α and α_{df} as the hyperparameters. As mentioned in the last section, "default" means the label for this document is unknown or unsure (in other words, not within S). The difference between α and α_{df} should reflect how conservatively we choose between regular topics/labels and the special "default" one.

Initially, all the documents inside the reference knowledge base (e.g. Wikipedia) are labeled documents, while all the documents in the external corpus are unlabeled documents. For each word w in both labeled and unlabeled documents, its label z is either the same as the label of the document in which it appears, or the special "background" label. The selection is controlled by an indicator variable t drawn from a Bernoulli distribution μ, which itself is drawn from a Beta prior with γ_1 and γ_2 as the hyperparameters. The difference between γ_1 and γ_2 should reflect the proportion of background topic. For each label in $S \cup$ "*default*" $\cup$ "*background*", it is associated with a multinomial distribution ϕ over words, which is drawn from the Dirichlet prior with β, β_{bg} and β_{df} as the hyperparameters. The difference among β, β_{bg} and β_{df} should reflect the content difference among regular labels, the "default" label and the "background" label. 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 y and the label-word association ϕ from this model. The document-label association helps reveal the entity labels for unlabeled documents, and the label-word association helps demonstrate the disambiguation evidences for each referent entity candidate.

To summarize, the detailed generative process of our model is as follows:

1. Draw the multinomial distribution over words $\phi_k \sim$ *Dirichlet*(β) for each regular topic/label k.

2. Draw the multinomial distribution over words $\phi_{bg} \sim$ *Dirichlet*(β_{bg}) for the background topic/label.

3. Draw the multinomial distribution over words $\phi_{df} \sim$ *Dirichlet*(β_{df}) for the default topic/label.

4. Draw a topic/label distribution $\theta \sim$ *Dirichlet*$(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = (\alpha_{df}, \alpha_1, ..., \alpha_k)$ and $\alpha_1 = \ldots = \alpha_k = \alpha$.

5. For each document $d \in D$:

   (a) Choose a topic/label $y_d \sim$ *Multinomial*(θ).

   (b) Choose a background topic proportion $\mu_d \sim$ *Beta*(γ_1, γ_2).

   (c) For each word position i in document d:

      i. Choose a background indicator $t_{di} \sim$ *Bernoulli*(μ_d).

      ii. if $t_{di} = 0$:

         A. Choose topic/label $z_{di} = bg$.

      iii. else:

         A. Choose topic/label $z_{di} = y_d$.

      iv. Choose a word $w_{di} \sim$ *Multinomial*$(\phi_{z_{di}})$.

Note that the above generative process is for unlabeled documents. For labeled documents, the document label y_d is known and fixed. Thus 5(a) becomes unnecessary and should be skipped. The other steps will remain the same.

Compared with regular topic models like LDA [3], our model is different in three aspects:

1. In our model, the regular topics, the "default" topic and the "background" topic may have multinomial distributions over words from different Dirichlet priors; while in LDA, the multinomial distributions over words are generated from the same Dirichlet prior.

2. In our model, each document has only one topic/label since we assume that the document is centered around the entity mention and the mention refers to a single entity; while in LDA, each document is a mixture of different topics.

3. In our model, a word can only have two possible labels: foreground or background, and the foreground label is restricted by the document label; while in LDA, the word topic is generated directly from a multinomial distribution over topics.

3.3 Incremental Evidence Mining Algorithm

Now we will explain our incremental evidence mining algorithm based on the model introduced in the last section. As discussed in Sections 3.1 and 3.2, our model is able to infer both the document-label association and the label-word association (the inference details will be discussed in next section). So after a run of our model, each unlabeled document will be assigned a label with the maximum likelihood (Section 3.4.3), and the words associated with each label will change accordingly (Section 3.4.4). Each run of the model will bring in some new knowledge (e.g. more labeled documents and more comprehensive label-word correspondences) and those new knowledge can further help the model to find more additional evidences. So this is a typical incremental mining scenario. We thus introduce an incremental evidence mining algorithm, as described in Algorithm 1. Algorithm 1 will first do inference only for the labeled documents of named entity mention m in reference knowledge base K (e.g. documents which contain mention m and a hyperlink to m's real referent entity). Then in each iteration, the algorithm will collect additional documents (D_{add}) from an external corpus C which have overlapped words with the current labeled documents (D_{i-1}). Inference will then be performed on current documents (D) and newly added documents (D_{add}) together, with a constraint that the labels of the current labeled documents (D_{i-1}) will remain unchanged. After inference, documents whose labels are found in the knowledge base will be added into the new labeled document set (D_i). The incremental process will continue until the iteration limit ($MaxIter$) is reached.

Algorithm 1 Incremental Evidence Mining

```
Input: Reference knowledge base $K$, external corpus $C$, named entity mention $m$, integer $MaxIter$.
$D_0 \leftarrow$ the set of labeled documents for mention $m$ in $K$
$S \leftarrow$ the set of entity candidates' labels for mention $m$
$D \leftarrow D_0$
Do inference for $D = D_0$
for all $i$ from 1 to $MaxIter$ do
    $D_{add} \leftarrow$ the set of documents in $C - D$ which have overlapped words with $D_{i-1}$
    $D \leftarrow D \cup D_{add}$
    Do inference for $D$ but fix labels for documents in $D_{i-1}$
    $D_i \leftarrow$ the set of documents in $D$ whose labels are in $S$
end for
```

3.4 Inference Algorithm

3.4.1 Likelihood Functions

The joint likelihood is

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

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

We use $\Gamma = \{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}\}$ to denote the hyperparameters. We would like to calculate the posterior probability of $p(\boldsymbol{t},\boldsymbol{y},\boldsymbol{z}|\boldsymbol{w},\Gamma)$ and use the maximal marginal probability to infer each topic assignment y_d and z_{di}. In typical topic models with conjugate prior such as LDA, one can apply collapsed Gibbs sampling to iteratively sample the variables z_{di}, t_{di}, y_d one by one, and estimate marginal probabilities with the samples. However, in our model, it is difficult to apply that sampling method due to the fact that every document has only one label y_d. In fact, when y_d and t_{di} are determined, z_{di} is uniquely decided as either bg or y_d. Therefore, if we sample y_d and z_{di} alternatively, once y_d is assigned some value fg, all the z_{di}'s associated with the corresponding document can only take values from $\{fg, bg\}$, and hence y_d will never be assigned any other value than fg because $p(y_d = l|z_{di} \in \{fg, bg\}) = 0$ for any $l \neq fg$. In other words, the Gibbs sampler will be trapped in a particular region $y_d = fg$ and never able to jump out of it.

To overcome that issue, we propose a blocked and collapsed Gibbs sampling algorithm with variational approximation.

3.4.2 A Blocked and Collapsed Gibbs Sampler with Variational Approximation

A blocked Gibbs sampler groups two or more variables together and samples from their joint distribution conditioned on all other variables, rather than sampling from each one individually. In our model, we group the variables $z_{d\cdot}, t_{d\cdot}$ and y_d for the same document together because of the aforementioned "deterministic trap" issue. In each blocked sampling stage, we sample the variables $z_{d\cdot}, t_{d\cdot}, y_d$ for one document d with all the other variables fixed as given.

Algorithm 2 Blocked Gibbs Sampling

```
for all $iter$ from 1 to $MaxIter$ do
    for all $d \in D$ do
        sample $\{z_{d\cdot}, t_{d\cdot}, y_d\}$ together according to $p(z_{d\cdot}, t_{d\cdot}, y_d|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}_{-d}, \Gamma)$ //call Algorithm 3
    end for
end for
```

Now we explain how we sample $z_{d\cdot}, t_{d\cdot}, y_d$. First, we notice that z_{di} is determined by t_{di} and y_d, so we only need to sample $t_{d\cdot}$ and y_d according to $p(t_{d\cdot}, y_d|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}_{-d}, \Gamma)$. Second, based on chain rule of joint probability we have:

$$p(t_{d\cdot}, y_d|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}_{-d}, \Gamma) = p(y_d|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}_{-d}, \Gamma)$$
$$\prod_i p(t_{di}|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma, t_{d1} \ldots t_{di-1})$$

So for a particular document d, we can first sample y_d and then sample t_{di} for each position i in d.

However, it is hard to compute $p(y_d|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}_{-d}, \Gamma)$ exactly because the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\mu}$ are hard to be integrated out when marginalizing $p(w_{d\cdot}, t_{d\cdot}|\boldsymbol{w}_{-d}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma)$. To sample y_d with the advantage of collapsed sampler, we make a variational approximation

$$p(t_{d\cdot}|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma) = \prod_i \psi(t_{di}|w_{di}, \boldsymbol{w}_{-d}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma)$$

where $\psi(t_{di}|w_{di}, \boldsymbol{w}_{-d}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma)$ is a variational distribution of $p(t_{di}|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma)$ as if d has only one word w_{di}. In other words, we temporarily assume the labels $t_{d\cdot}$ in one document are conditionally independent given y_d and all variables in other documents. This approximation is reasonable because our documents are short but the number of documents is large. The conditional probability of $p(t_{d\cdot}|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma)$ changes little with this approximation but the calculation of $p(y_d|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}_{-d}, \Gamma)$ now becomes easy to accomplish[2].

$$p(y_d = l|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}_{-d}, \Gamma) \propto \frac{\alpha_l + |\{y_{d'} = l, d \neq d'\}|}{\sum_{k=1}^{K} \alpha_k + \alpha_{df} + |D| - 1} \prod_{i=1}^{N_d}$$
$$\left(\frac{\gamma_1 + |\{t_{d'j} = 0, d' \neq d\}|}{\gamma_1 + \gamma_2 + \sum_{d' \neq d} N_{d'}} \frac{\beta_{bg} + |\{t_{d'j} = 0, w_{d'j} = w_{di}, d' \neq d\}|}{|W|\beta_{bg} + |\{t_{d'j} = 0, d' \neq d\}|} \right.$$
$$\left. + \frac{\gamma_2 + |\{t_{d'j} = 1, d' \neq d\}|}{\gamma_1 + \gamma_2 + \sum_{d' \neq d} N_{d'}} \frac{\beta_l + |\{z_{d'j} = l, w_{d'j} = w_{di}, d' \neq d\}|}{|W|\beta_l + |\{z_{d'j} = l, d' \neq d\}|} \right) \tag{2}$$

After we sample y_d, we can sample t_{di} for each position i in d. If $y_d = default$,

$$\frac{p(t_{di} = 0|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma)}{p(t_{di} = 1|\boldsymbol{w}, \boldsymbol{z}_{-d}, \boldsymbol{t}_{-d}, \boldsymbol{y}, \Gamma)} = \frac{\gamma_1 + |\{t_{d'j} = 0, d' \neq d\}|}{\gamma_2 + |\{t_{d'j} = 1, d' \neq d\}|}$$
$$\cdot \frac{\beta_{bg} + |\{t_{d'j} = 0, w_{d'j} = w_{di}, d' \neq d\}|}{\beta_{df} + |\{z_{d'j} = y_d, w_{d'j} = w_{di}, d' \neq d\}|}$$
$$\cdot \frac{|W|\beta_{df} + |\{z_{d'j} = y_d, d' \neq d\}|}{|W|\beta_{bg} + |\{t_{d'j} = 0, d' \neq d\}|} \tag{3}$$

Otherwise replace the β_{df} in the formula with β. Finally, we have the following sampling steps for sampling one block.

Algorithm 3 The subroutine for sampling One Block (Line 3 in Algorithm 2)

```
Sample $y_d$ according to Eq. (2)
for all $i$ from 1 to $N_d$ do
  Sample $t_{di}$ according to Eq. (3)
  if $t_{di} = 0$ then
    $z_{di} \leftarrow bg$
  else
    $z_{di} \leftarrow y_d$
  end if
end for
```

3.4.3 Estimating Document Label

We infer the document label using maximal marginal probability with one exception: if the maximal marginal probability is smaller than a threshold η, we predict the label to be $default$. With this threshold we can control the noise by only labeling the documents on which our model has sufficiently high confidence.

$$y_d = \begin{cases} \arg\max_k p(y_d = k|\boldsymbol{w}, \Gamma) & \max_k p(y_d = k|\boldsymbol{w}, \Gamma) \geq \eta \\ default & \max_k p(y_d = k|\boldsymbol{w}, \Gamma) < \eta \end{cases} \tag{4}$$

Since our model can infer the document label for unlabeled document, it can also be directly used for named entity disambiguation if we treat the query as an unlabeled document.

3.4.4 Estimating Label-Word Association

We infer the label of each word in each document with maximal marginal probability:

$$t_{di} = \arg\max_{l \in \{0,1\}} p(t_{di} = l|\boldsymbol{w}, \Gamma) \tag{5}$$

$$z_{di} = \begin{cases} y_d & t_{di} = 1 \\ default & t_{di} = 0 \end{cases} \tag{6}$$

And the label-word distribution can be estimated by *maximum a posteriori* (MAP) inference:

$$\phi_k^{(v)} = \frac{\beta_k + |\{z_{di} = k, w_{di} = v\}|}{|W|\beta_k + |\{z_{di} = k\}|} \tag{7}$$

4. EXPERIMENTS

In this section, we evaluate the effectiveness of our proposed method for the MENED task on two real-life datasets: one from news, and the other from Twitter. We will: (1) compare the disambiguation accuracy and robustness of our method, to two state-of-the-art NED methods that utilize various kinds of features; (2) analyze the effectiveness of the additional evidences mined by our method; (3) show how the performance of our method changes with respect to the number of the incremental evidence mining iterations. All the experiments, if not specifically mentioned, are conducted on a server with 2.40GHz Intel Xeon CPU and 48GB RAM.

4.1 Datasets

Since our work is the first one to tackle the MENED problem, there is no established publicly available benchmark for us to test. As mentioned before, the goal of MENED is to bridge the potential information gap between a query and the reference knowledge base. Here we use two real-world datasets where such information gap indeed exists, to test the performance of our algorithm.

The first one is derived from the TAC-KBP2009 dataset, which is created for the Entity Linking task [20] in the Knowledge Base Population track at the Text Analysis Conference. The TAC-KBP2009 dataset consists of 3,904 queries (again, each query is an entity mention along with its context) and entity mentions in 1,675 of them can be linked to their corresponding entries in the knowledge base (Wikipedia). The fact that more than half of the entity mentions cannot find their underlying entities in the knowledge base also proves that the reference knowledge base is usually a limited information source and therefore it may lack some important information to help disambiguate named entity mentions. The original dataset contains many long news articles. In order to test the abilities of different algorithms in a challenging scenario where information gap is large, we modify this dataset to keep only a fixed-size word window surrounding the query mention as its "context" (in this work, we choose the word window size as 60). By adding this constraint the information gap is enlarged and the disambiguation difficulty is increased. Among the 1,675 resolvable queries, we choose the queries whose named entity mentions have a corresponding Disambiguation Page in Wikipedia as our first test dataset. This dataset contains 424 queries.

Our second dataset is generated from Twitter. Since tweets have the 140-character constraint and the words used in them are often irregular, the probability of seeing information gap between tweets and the reference knowledge base is relatively high. Therefore running NED on tweets is much harder than on news. We randomly picked 25 ambiguous

[2] This approximation can be avoided if we use a variable elimination trick. However, with this approximation the sampling is more efficient.

entities from the Wikipedia's Disambiguation Page Category and crawled 500 tweets containing these mentions as queries. After filtering out the queries which are unsolvable (e.g. even human beings cannot specify which entity the mention refers to), 340 queries are left and we treat them as our second test dataset.

4.2 Experiments Setup

In this work, we use Wikipedia as the reference knowledge base and the webpages indexed by Google as the external corpus. For each reference entity candidate, we generate its labeled data (D_0 in Section 3.3) by utilizing its Wikipedia page and all Wikipedia pages which have hyperlinks to its Wikipedia page. For fetching related documents (D_{add} in Section 3.3) from the external corpus, we make use of the Google Search API and collect the top 20 webpages for each referent entity candidate.

4.3 NED Accuracy & Robustness

We first conduct experiments to compare our method with two NED methods utilizing various kinds of features: **Wikifier** [24], a state-of-the-art NED system using a machine learning based hybrid strategy to combine popularity prior, context similarity and topical coherence features together, and **AIDA** [19], a robust NED system making use of weighted mention-entity graph to find the best joint mention-entity mapping. As explained in Section 3.4.3, our model for MENED can be directly used for NED if we treat the query as an unlabeled document. We test our model under two settings: (1) using the evidences mined from Wikipedia only; (2) using the evidences mined from both Wikipedia and the external corpus. We denote the first setting as **MENED(Wiki)** and the second setting as **MENED(All)**. For both MENED(Wiki) and MENED(All), we use the following parameter settings: $\alpha = 0.001, \alpha_{df} = 0.01, \beta = 0.001, \beta_{df} = 0.01, \beta_{bg} = 0.1, \gamma_1 = 0.0003, \gamma_2 = 0.001$. These parameters are tuned on a small test dataset containing 15 queries and then reused in all the experiments without any further tuning. The threshold for predicting document label is chosen as $\eta = 0.9$. For MENED (All), we incrementally mine evidences from external corpus for 5 rounds. For Wikifier and AIDA, we use the parameter settings suggested by their authors. The same parameter settings were applied to both datasets. Wikifier used a Wikipedia repository of 2009[3]. Originally the Wikipedia repository used by AIDA is of 2010, later the authors kindly provided us an updated version which used a Wikipedia repository of late 2012. We denote the original one and the updated one as AIDA(2010) and AIDA(2012), respectively. Both MENED(Wiki) and MENED(All) rely on a Wikipedia repository of late 2012.

Figure 4 shows that MENED(All) slightly outperforms Wikifier and AIDA on TAC-KBP2009 dataset. Compared with Wikifier and AIDA, MENED(All) does not utilize any complicated features (e.g. topical coherence). On the Twitter dataset, MENED(All) performs remarkably better than Wikifier and AIDA. Both Wikifier and AIDA get very poor NED accuracy on short and noisy texts like tweets. MENED(All) retains high accuracy on tweets, indicating a much more robust performance. We notice that MENED(Wiki) also greatly outperforms Wikifier and AIDA on the Twitter dataset. This is due to two reasons. First, MENED(Wiki) mines new evidences from the Wikipedia pages that hyperlink to the entity candidates' own Wikipedia pages. Second, the topical coherence feature utilized by both Wikifier and AIDA is less helpful on very short texts like tweets since there are very few entities within the short context.

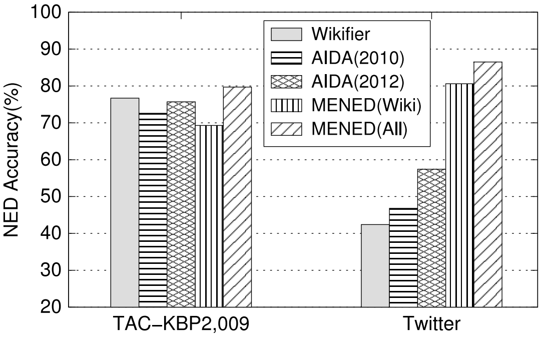

Figure 4: MENED vs. Wikifier vs. AIDA

We also tried to compare our method with **TAGME** [13], an NED system specifically designed for very short texts like tweets. Different from Wikifier and AIDA, the TAGME API does not allow us to specify the named entity mentions to disambiguate. As a result, many queries in our test datasets are not properly responded. Without considering the queries which cannot be handled, TAGME obtains the NED accuracy of 78.3% and 61.1% on TAC-KBP2009 data and Twitter data respectively. Our method MENED(All) outperforms TAGME on both datasets.

4.4 Effectiveness of Evidence Mining

We then conduct experiments to demonstrate the effectiveness of mining evidences from external corpus. As can be seen from Figure 4, MENED(All) outperforms MENED(Wiki) in terms of NED accuracy on both datasets. The accuracy gain illustrates that our method for MENED is effective and the mined evidences from external corpus are indeed very helpful for boosting the NED performance.

Table 2 shows the mined evidences from external corpus for several entities. We can see that the mined evidences can provide complementary knowledge for disambiguating the entities, especially for those entities that are not very popular and therefore do not have many context information in Wikipedia. For example, in "Michael I. Jordan" case, the evidences "layers, nonparametric, nonlinear" correspond to his research work, "pehong, chen, distinguished" indicate the fact that he is a Pehong Chen Distinguished Professor at UC Berkeley, and "david, heckerman, kearns, marina, meila" describe his collaborators. All these evidences are not captured in Wikipedia but available in external sources (e.g. his homepage and DBLP page). Our model and algorithm can successfully dig out these useful evidences scattered across multiple documents in the external corpus.

4.5 Impact of Evidence Mining Iterations

Next we conduct experiments to illustrate how the performance of our MENED method changes with respect to the number of incremental evidence mining iterations. Here the parameter settings are the same as those described in Section 4.3. Figure 5 shows that the NED accuracy increases as

[3] The authors of Wikifier are working on a updated version utilizing recent Wikipedia repository, but unfortunately it cannot be ready at this time. We were also unable to obtain a Wikipedia repository of 2009 to run our method with.

Entity	Mined Additional Evidences
Michael I. Jordan (Michael Jordan)	layers, nonparametric, non-linear, pehong, chen, distinguished, david, heckerman, kearns, marina, meila ...
Michael B. Jordan (Michael Jordan)	wood, oscar, role, peters, gilliard, detmer, larry, freamon, true-frost, pryzbylewski, octavia, spencer, troubled, ...
Owen Bieber (Bieber)	jobs, automobile, corporation, approved, presidential, lofton, support, vote, organizer, worley, conventions, worker ...
General Aircraft Hotspur (Hotspur)	operating, ground, states, cargo, aviation, capacity, built, fighter, targets, spitfire, flight, eben, paratroops ...
David Young Cameron (David Cameron)	engravers, technique, sculpture, printmaking, reproduced, scotch, lorne, muirhead, walton, french, nature, lovely ...

Table 2: Mined Evidences for Michael I. Jordan, Michael B. Jordan, Owen Bieber, General Aircraft Hotspur and David Young Cameron. Words in parentheses are named entity mentions.

the number of iterations increases. But the increasing speed slows down as more evidences are collected.

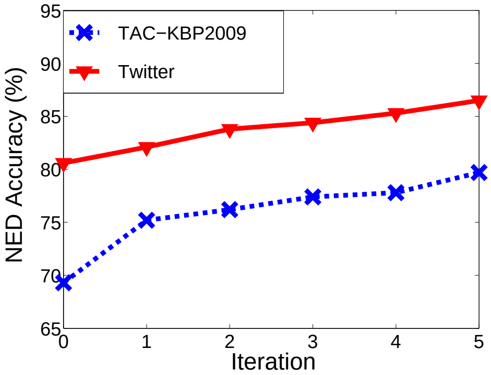

Figure 5: Varying Evidence Mining Iterations

5. RELATED WORK

Named entity disambiguation has received a lot of attentions in recent years. Approaches that disambiguate named entity mentions with respect to Wikipedia date back to Bunescu and Pasca's work [5]. 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. Several subsequent work incorporated more information into similarity comparison: Gottipati and Jiang [14] explored query expansion, while Zhang and Sim [28] considered acronym expansion. To incorporate different types of disambiguation knowledge together, Han and Sun [16] proposed a generative model to include evidences 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 [25] adopted a latent topic model to learn the context-entity association to help disambiguation. Cucerzan's work [10] is the first one to realize the effectiveness of using topical coherence to help named entity disambiguation. In that work, the topical coherence 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 and Witten [22] refined Cucerzan's work by defining topical coherence using Normalized Google Distance [9] and only using "unambiguous entities" in the context to calculate topical coherence. Several new measures of topical coherence were also proposed in recent years: Bhattacharya and Gatoor [2] modeled the topical coherence as the association of an entity and the latent topics of a document, and Sen [25] modeled the topical coherence using the co-occurrence of entities. Recently, several methods [18, 19, 24, 26, 28] also tried to combine together "context similarity" and "topical coherence" using a hybrid strategy which could further improve disambiguation accuracy.

Almost all these previous NED algorithms fall into the scope of "single document NED". Their disambiguation decisions depend on the comparison (both textual and topical) of the query document (named entity mention along with its context) and the referent entity candidates' Wikipedia pages. Therefore they cannot handle the cases where there are no enough overlaps between the compared documents. To solve this problem, Chen and Ji [8] proposed the "Collaborative Ranking" technique. In their work, they used document clustering to find several "query collaborators" (documents which are in the same cluster with the query document) and ran existing NED algorithms on each "query collaborator" separately. Their disambiguation results were then assembled together to make the final decision. If most of a query's collaborators exhibit enough overlap with referent entity candidates' Wikipedia pages, the final disambiguation decision will likely be reasonable. Han and Sun [17] tackled this problem in another way. They proposed a generative entity-topic model which can jointly model context compatibility, topical coherence and their correlations. Since their model was trained on all Wikipedia pages, it made use of not only the contents of referent entity candidates' Wikipedia pages, but also the contents of the Wikipedia pages where those referent entity candidates appear. Compared with "single document NED" algorithms, their method utilized cross-document information. However, both [8] and [17] cannot work well in cases where a query's context information does not exist in the entire Wikipedia corpus. Our work focuses on such cases and explores information both inside and outside Wikipedia to mine additional evidences for named entity disambiguation.

Our proposed model is inherited from the Latent Dirichlet Allocation (LDA) model. LDA was first proposed by Blei, Ng and Jordan [3] for finding the document-topic association and the topic-word association in text documents. Ramage, Hall, Nallapati and Manning [23] 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 entity disambiguation evidence mining, and our semi-supervised learning model works in an incremental manner to control errors. Most LDA-based models require a preprocessing step to remove the stopwords. Otherwise, those stopwords will pervade the learnt topics, hiding the real statistically interesting word patterns. However, removing stopwords is not trivial as a lot of stopwords are domain-dependent. To cope with this issue, several work [7, 15] introduced a special background topic and assumed all stopwords being generated by this background distribution. Our model incorporates not only such a background distribution to capture stopwords, but also a "default" distribution to capture words from documents with unknown or unsure labels. Therefore our model is robust and effective in mining evidences for named entity disambiguation.

6. CONCLUSIONS

In this paper, we studied the problem of mining evidences for named entity disambiguation. We proposed a generative model and an incremental algorithm to automatically mine useful evidences across documents. With a specific modeling of "background topic" and "unknown entities", our model is able to harvest useful evidences from noisy text. To evaluate the effectiveness of our model and algorithm, a thorough experimental study was conducted. The experimental results demonstrated that our proposed method can mine additional evidences to significantly boost the disambiguation performance. As future work, we plan to extend our approach to mine other type of evidences such as entities and concepts. We would also like to combine the evidences mined by our method with other NED algorithms.

7. ACKNOWLEDGMENTS

This research was sponsored in part by the Army Research Laboratory under cooperative agreements W911NF-09-2-0053, NSF IIS-0954125, and DARPA under agreement number FA8750-13-2-0008. Chi Wang was supported by Microsoft Graduate Fellowship. 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.

8. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. *The Semantic Web*, pages 722–735, 2007.

[2] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. pages 509–518, 2006.

[3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*, pages 1247–1250, 2008.

[5] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, pages 9–16, 2006.

[6] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka Jr, and T. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of AAAI*, pages 1306–1313, 2010.

[7] C. Chemudugunta and P. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *Proceedings of NIPS*, pages 241–248, 2007.

[8] Z. Chen and H. Ji. Collaborative ranking: A case study on entity linking. In *Proceedings of EMNLP*, pages 771–781, 2011.

[9] R. Cilibrasi and P. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.

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

[11] 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.

[12] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of EMNLP*, pages 1535–1545, 2011.

[13] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of CIKM*, pages 1625–1628, 2010.

[14] S. Gottipati and J. Jiang. Linking entities to a knowledge base with query expansion. In *Proceedings of EMNLP*, pages 804–813, 2011.

[15] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of ACL-HLT*, pages 362–370, 2009.

[16] 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.

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

[18] 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.

[19] 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.

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

[21] 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.

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

[23] 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.

[24] 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.

[25] P. Sen. Collective context-aware topic models for entity disambiguation. In *Proceedings of WWW*, pages 729–738, 2012.

[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] F. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of WWW*, pages 697–706, 2007.

[28] 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.