Targeted Disambiguation of Ad-hoc, Homogeneous Sets of Named Entities

Chi Wang [*]
University of Illinois at Urbana-Champaign
Champaign, IL
chiwang1@illinois.edu

Kaushik Chakrabarti, Tao Cheng,
Surajit Chaudhuri
Microsoft Research, Redmond, WA
{kaushik,taocheng,surajitc}@microsoft.com

ABSTRACT

In many entity extraction applications, the entities to be recognized are constrained to be from a list of "target entities". In many cases, these target entities are (i) ad-hoc, i.e., do not exist in a knowledge base and (ii) homogeneous (e.g., all the entities are IT companies). We study the following novel disambiguation problem in this unique setting: given the candidate mentions of all the target entities, determine which ones are true mentions of a target entity. Prior techniques only consider target entities present in a knowledge base and/or having a rich set of attributes. In this paper, we develop novel techniques that require no knowledge about the entities except their names. Our main insight is to leverage the homogeneity constraint and disambiguate the candidate mentions collectively across all documents. We propose a graph-based model, called MentionRank, for that purpose. Furthermore, if additional knowledge is available for some or all of the entities, our model can leverage it to further improve quality. Our experiments demonstrate the effectiveness of our model. To the best of our knowledge, this is the first work on targeted entity disambiguation for ad-hoc entities.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Named Entity Disambiguation, Entity Extraction, Targeted Disambiguation, Ad-hoc Entity Identification, MentionRank

1. INTRODUCTION

Many applications need to identify mentions of named entities in text documents. Consider the Voice of the Customer (VoC) application. Here, the enterprise is typically interested in mining customer sentiment of its own and its competitor's products. It maintains a list of entities and requires identification of mentions of *only these entities* in the web documents. We refer to them as *target entities*. These target entities have two unique characteristics:

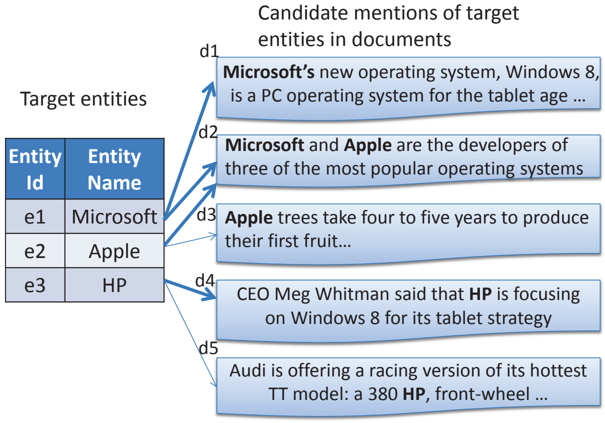

Candidate mentions of target entities in documents
Target entities
Entity Id
Entity Name
e1
Microsoft
e2
Apple
e3
HP
d1
Microsoft's new operating system, Windows 8, is a PC operating system for the tablet age …
d2
Microsoft and **Apple** are the developers of three of the most popular operating systems
d3
Apple trees take four to five years to produce their first fruit…
d4
CEO Meg Whitman said that **HP** is focusing on Windows 8 for its tablet strategy
d5
Audi is offering a racing version of its hottest TT model: a 380 **HP**, front-wheel …

Figure 1: Targeted disambiguation problem. Candidate mentions are shown using arrows; true mentions shown using thick arrows.

- **Ad-hoc**: All or many of the target entities are *ad-hoc*, i.e., they do not exist in a knowledge base such as DBpedia, Freebase or YAGO. Consider a VoC application mining sentiment of various brands of shoes: there are more than 900 different active brands [1] but only 82 exist in Wikipedia. This issue is much more pronounced for entities from "tail" domains. For such entities, *all we have are their names.*
- **Homogeneous**: Many applications perform analyses for one *homogeneous* group of entities at a time; for example, a VoC application typically mines the sentiment for shoe brands separately from that for shirt brands. Supposing the target entities are present in an "is-a" ontology like YAGO, we deem the target entities to be homogeneous if their least common ancestor (LCA) is far from the root. The LCA of the target entities in the is-a ontology is referred to as the *"target domain"*. Note that we do not require the entities to belong to any existing ontology; we use this notion just to introduce the concept of homogeneity.

Names (surface forms) of entities are often ambiguous and can have many different meanings. So, the name of a target entity appearing in a document does not necessarily mean it refers to the target entity. Consider the target entities of the domain "IT companies" in Figure 1. The string "Apple" appears in documents $d2$ and $d3$ but only $d2$ refers to the IT company Apple. We refer to the former as "candidate mentions" of the target entity and the latter as its "true

[*] Work was done while the author was in Microsoft Research.

[1] According to shoes.com

WWW 2012, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1229-5/12/04.

mentions". Identifying the true mentions amongst all the candidate mentions is crucial for many applications.

We study the following novel disambiguation problem in this unique setting: *given the candidate mentions of all the target entities, determine which ones are true mentions of the target domain*. We refer to it as *targeted entity disambiguation* (TED). In Figure 1, TED should output the candidate mentions in $d1$, $d2$ and $d4$ as true mentions and those in $d3$ and $d5$ as false mentions. In this paper, when we refer to disambiguation, we imply TED.

Prior work and limitations: Previous "entity linking" techniques mostly focus on entities present in a knowledge base like DBpedia or YAGO [5, 9, 17, 18, 8, 15, 14, 12]. The main idea is to consider the "context" (i.e., surrounding words) of the candidate mention and compare it, by some similarity measure, to text metadata associated with the entity in the knowledge base. We refer to this general approach as *ContentMatch* approach in this paper.

The above approaches have the following limitations. First, they are not effective when some or all the entities are adhoc; this is confirmed by our experiments. Second, even if all the entities are present in a knowledge base, these approaches are biased towards pages that are similar to the entity metadata in the knowledge base (e.g., content of their Wikipedia page). For short and informal documents (e.g., tweets, forum postings) this may result in poor quality.

Main insights and contributions: A good solution to TED should (i) require no knowledge about the entities except the names of the entities and (ii) be able to leverage additional knowledge (e.g., Wikipedia page, rich set of attributes) to improve the quality if it is available for some or all of the entities. We ask the question: although the adhocness makes the problem more challenging, can we leverage the homogeneity constraint to solve this problem? Our main insight is to leverage the homogeneity contraint of the entities in the following three ways.

- *Context similarity*: The true mentions *across all the entities across all the documents* will have more similar contexts than the false mentions of different entities. This is because the true mentions refer to entities in the same domain while the false mentions point to things in disparate domains. For example, in Figure 1, the mention of $e1$ in $d1$ (denoted by $(e1, d1)$) and mentions $(e1, d2)$ and $(e2, d2)$ (true mentions) have similar contexts (e.g., common words: "operating", "system"). On the other hand, the false mentions $(e2, d3)$ and $(e3, d5)$ do not have context similar with each other or with the true mentions. A crucial point is that the false mentions *for any individual entity* can have similar contexts among them (e.g., among the mentions of the fruit "Apple") but it *does not span across entities* (e.g., between the mentions of the fruit "Apple" and the mentions of power unit HP).
- *Co-mention*: If multiple target entities are co-mentioned in a document, they are likely to be true mentions. For example, $d2$ co-mentions Microsoft and Apple, hence they are very likely to be true mentions of those entities. Note that our insight on co-mention is different from that of coherence used in [15, 14]; they require the entities to exist in a knowledge base to compute the coherence whereas we do not.
- *Cross-document, cross-entity interdependence*: If one or more mentions among the ones with similar context is deemed likely to be a true mention (based on some evidence), they are all likely to be true mentions. For example, since $(e1, d2)$ and $(e2, d2)$ are deemed likely to be a true mention due to co-mention, $(e1, d1)$ is also likely to be a true mention as its context is similar to the former's. Consequently, $(e3, d4)$ is likely to be a true mention as it is similar to $(e1, d1)$ in terms of context.

This gives rise to several technical challenges. How do we leverage the above insights? Is it possible to unify them? If additional knowledge is available for some of the entities, how can we leverage it in conjunction with these insights?

Our main contributions can be summarized as follows:

- We propose a novel graph-based model, called *MentionRank*, that unifies the above three insights and disambiguates all the mentions across all the documents collectively. It can perform *targeted disambiguation without any knowledge about the entities besides their names*. To the best of our knowledge, our approach is the first with this ability.
- If additional knowledge is available for some or all the entities (e.g., reference page for the entity from a knowledge base), we can incorporate it into our MentionRank model. Due to cross-document, cross-entity interdependence, authentic documents not only improve disambiguation of the entities they correspond to but also other target entities.
- We perform extensive experiments on three real-life entity sets and a random sample of the web documents. Our experiments demonstrate (i) without any authentic document for any entity, MentionRank can achieve similar quality as ContentMatch that must use authentic documents and (ii) with authentic documents for some entities, MentionRank outperforms ContentMatch techniques.

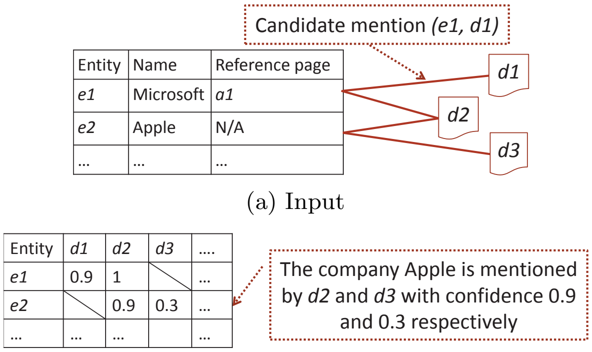
2. TARGETED DISAMBIGUATION PROBLEM

In our problem setting, we take (i) a set of entity names, (ii) a collection of documents, and (iii) candidate mentions of the given entities in the given documents as input, and determine which ones are true mentions. The uniqueness of this problem is that the entities are all in the same domain (referred to as the target domain). Note that we do not require the user to specify what the domain is.

Candidate mentions are obtained by a separate component that identifies occurrences of the entity names in the documents. An entity name can appear in a document multiple times. Following standard practice [10], we assume that all occurrences of a name inside a document refer to the same entity (*e.g.*, occurrences of the string "Apple" in a single document either all refer to the IT company or all refer to the fruit). So we perform disambiguation for each entity-document pair where the entity name has occurred, rather than disambiguating each single occurrence of the entity name. From now on, we use *mention* to refer to such a entity-document coupling, and *occurrence* to refer to a specific occurrence.

The same entity can have multiple name variants, like HP and Hewlett-Packard. If the variants are known (e.g., using a entity synonym module [7]), the candidate mention identification component will identify more candidate mentions of each entity. However, this is orthogonal to our problem because our problem takes the candidate mentions as input.

For purpose of flexibility, the system produces a score between 0 and 1 for each candidate mention to indicate the likelihood of it being a true mention. The scores are globally comparable across all entities. The first benefit is a user can



Entity	Name	Reference page
e1	Microsoft	*a1*
e2	Apple	N/A
...	...	...

Candidate mention *(e1, d1)*
d1
d2
d3

(a) Input

Entity	*d1*	*d2*	*d3*	....
e1	0.9	1		...
e2		0.9	0.3	...
...	...	...		...

The company Apple is mentioned by *d2* and *d3* with confidence 0.9 and 0.3 respectively

(b) Output

Figure 2: Examples for the input and output.

decide the cut-off value for tradeoff between precision and recall. Second, users can retrieve top-k mentions per entity or per collection. For example, enterprise users may want to get a sufficient number of reviews for every product. On the other hand, a trader may want to get a few highly precise news about financial corporations in general (not necessarily for every financial company) in order to predict trading trends.

Formally we have the following problem definition.

Definition 1 (Targeted Entity Disambiguation). *Given input of a target entity set $E = \{e_1, \ldots, e_n\}$, a document set $D = \{d_1, \ldots, d_m\}$ and candidate mentions $R = \{(e_i, d_j)|e_i \in E, d_j \in D\}$, output score $r_{ij} \in [0, 1]$ for every candidate mention $(e_i, d_j) \in R$.*

Note that TED disambiguates *at the target domain level* and not at the target entity level. If the names of target entities are all distinct, the two are identical. Otherwise (e.g., two IT companies in the list named "Apple"), TED does not disambiguate among them. One can separate them in target domains of finer granularity, or apply existing techniques [16, 19] on the results of TED.

In an extended setting of TED in Section 4, we allow additional prior knowledge from users as input such as entity attributes and reference pages for some entities. A well-engineered system should work well even when the additional knowledge is incomplete or unavailable, and work better when it is available.

Example 1. *Figure 2 illustrates a running example of TED. Two entities, e_1 = Microsoft, e_2 = Apple and 3 documents containing them are shown, where $R = \{(e_1, d_1), (e_1, d_2), (e_2, d_2), (e_2, d_3), \ldots\}$. The reference page for e_1 is available, while the reference page for e_2 is unavailable. The output implies that $(e_1, d_1), (e_1, d_2), (e_2, d_2)$ are highly probable true mentions (indicated by high scores 0.9, 1.0 and 0.9 respectively), while the candidate mention (e_2, d_3) is not (indicated by low score 0.3).*

3. OUR APPROACH

3.1 Hypotheses

One unique property of our problem is that we aim to find the entity mentions in the same, albeit unknown, target domain. This has three different implications: the context between true mentions are similar within an entity as well as across different entities; the context of false mentions are not similar with true mentions; the context of false mentions can be similar among themselves within an entity, but dissimilar across different entities. The first two are easy to understand, as we assume the target entities are homogeneous. The last one is due to the observation that the meanings of false mentions (referred to as domain-external meanings) are usually not in the same domain. For example, "Apple", "HP", "Sun" all have meanings outside IT company domain – the fruit apple, the power unit HP(horse power), and many different meanings of Sun. The context of their domain-external meanings are dissimilar. This hypothesis is illustrated in Figure 3, and summarized as:

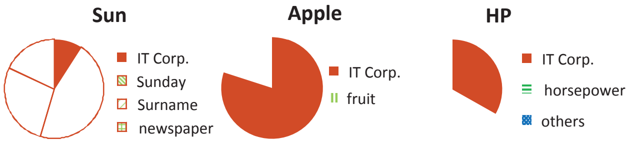


Figure 3: An illustration of context similarity hypothesis. Orange solid part is the similar context shared by mentions of the three IT companies; other parts are different domain-external meanings with different context.

Hypothesis 1 (Context Similarity). *The context between two true mentions is more similar than between two false mentions across two distinct entities, as well as between a true mention and a false mention.*

The second hypothesis is about the entity co-mention in one document.

Hypothesis 2 (Co-mention). *If multiple entity names in the given list are mentioned in the same document, the chance for them to refer to the entities in the target domain is higher.*

Finally, we have the hypothesis about the interdependency of disambiguation of different mentions.

Hypothesis 3 (Interdependency). *If a mention has similar context with many true mentions, it is likely a true mention.*

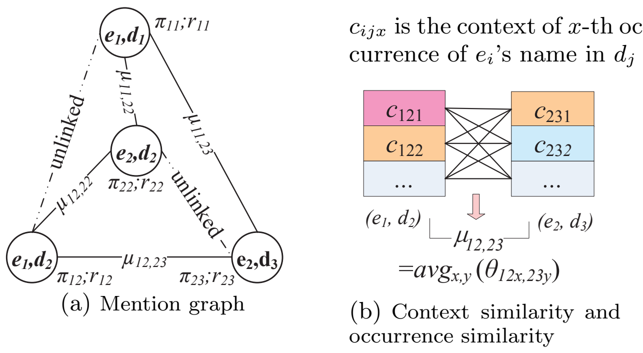
This resembles the philosophy of PageRank — one page that many popular pages point to may also be popular.

3.2 Graph-based model

To leverage the interdependencies of mention score, we model this problem with a graph-based ranking method. The disambiguation is performed by solving the ranking problem holistically.

Inspired by existing graph-based ranking methods such as PageRank, we build a *mention graph* and perform a PageRank-like ranking algorithm on it. We show an example in Figure 4(a). Each node represents a candidate mention (e_i, d_j). It is associated with a ranking score r_{ij}, as well as a prior estimation of the ranking score π_{ij}. Each edge links two interdependent mentions (e_i, d_j) and $(e_{i'}, d_{j'})$, and has a weight $\mu_{ij,i'j'}$, indicating how close the two ranking score r_{ij} and $r_{i'j'}$ should be. $\mathbf{r}$ is an unknown vector, while $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ are defined based on our hypotheses.

Context similarity as edge weight $\boldsymbol{\mu}$. According to the interdependency hypothesis, if two mentions have more similar context, the ranking score should be propagated in a



(a) Mention graph

(b) Context similarity and occurrence similarity

Figure 4: An illustration of the graph-based ranking model.

larger extent to each other. The edge weight is thus defined to be a similarity measure between the context of the two mentions. How the similarity is computed is described in Section 3.2.1.

Co-mention as prior $\boldsymbol{\pi}$. Due to the co-mention hypothesis, we define the prior estimation based on the degree of co-mentions. Several variations of it are described in Section 3.2.2.

The final score of each mention is decided by its prior estimation as well as the score of other correlated mentions. We use a parameter $\lambda \in [0, 1]$ to control the relative importance of the two parts.

$$r_{ij} = \lambda p_{ij} + (1 - \lambda) \sum_{i',j'} w_{ij,i'j'} r_{i'j'} \qquad (1)$$

where $\mathbf{w}$ is directed propagation weight obtained from the undirected edge weight $\boldsymbol{\mu}$, and $\mathbf{p}$ is the normalized prior estimation obtained from $\boldsymbol{\pi}$ such that $\sum_{i,j} p_{ij} = 1$. We will discuss the exact definition of $\mathbf{w}$ and $\mathbf{p}$ in following subsections. All hypotheses we discussed above have been organically integrated – the co-mention prior by $\mathbf{p}$, the context similarity by $\mathbf{w}$, and the interdependency by the product of $\mathbf{w}$ and $\mathbf{r}$. We refer to our model as MentionRank.

3.2.1 Context Similarity

We discuss how we compute the context similarity between two candidate mentions. Recall that a candidate mention is a (e_i, d_j) pair. Let s_i be the name of e_i. There can be multiple occurrences of s_i in d_j. The context similarity between two mentions can manifest itself in any pair of occurrences. So we compute similarity between all pairs and aggregate them, as shown in Figure 4(b). We define the context similarity between (e_i, d_j) and $(e_{i'}, d_{j'})$ as the average of the similarity between the context of every occurrence of s_i in d_j and $s_{i'}$ in $d_{j'}$.

$$\mu_{ij,i'j'} = \underset{x,y}{average}\, \theta_{ijx,i'j'y} \qquad (2)$$

where $\theta_{ijx,i'j'y}$ denotes the similarity between the context c_{ijx} of x-th occurrence of s_i in document d_j and the context $c_{i'j'y}$ of y-th occurrence of $s_{i'}$ in document $d_{j'}$; we refer to it as occurrence similarity. Alternative choices of context similarity are min, max, median or a mixture of these functions over the occurrence similarity. We found that the average is equal or slightly better than the others in most cases.

Now we define occurrence similarity. Following the practice of previous study [5, 9], we define the context c_{ijx} as a short snippet of l words before and l words after the x-th occurrence of s_i in document d_j. We use a simple but effective measure, tf-idf cosine similarity [11], as the basis of comparing two pieces of context. Other features and alternative similarity measures, like those summarized in [2], can also be used in our model.

We explore several variants of tf-idf vectors. For cosine similarity computation, we normalize each vector to have length 1, so that we only need to do dot product when computing similarity. To filter out noisy and undiscriminative words on the Web, we remove the words with very low frequency or very high document frequency in the corpus. We can either do the filtering first or do the normalization first. We empirically found that doing filtering after normalization is better.

3.2.2 Prior

Recall that we estimate the prior score for a candidate mention based on the co-mention degree with other input entities. The question is what to be considered as a "co-mention". We can consider the whole document as the text window to search for co-mentioned entities or restrict the co-mention to be within a short window. We explored the following two ways to define π_{ij}.

- number of unique names of target entities occurred in d_i.
- number of unique names of target entities occurred in the context of s_i in d_j, i.e., $\bigcup_x c_{ijx}$.

We found that the first way leads to better results. This may imply the long-range interaction of entities should be respected.

Once $\boldsymbol{\pi}$ is computed, the normalized prior $\boldsymbol{p}$ can be easily obtained: $p_{ij} = \pi_{ij} / \sum_{i,j} \pi_{ij}$.

3.2.3 Propagation Weight

We cannot directly use $\boldsymbol{\mu}$ as propagation weight because it is unnormalized raw similarity. Moreover, we need to design the propagation weight properly to address a "false-boost" issue discussed below.

Recall that we model the interdependency hypothesis with the weighted propagation along edges in the hope that a group of similar true mentions boost the ranking score of each other. One concern is that a group of similar false mentions will also boost their ranking score. This can happen when an entity has many domain-external mentions with similar context. For example, 99% of the Web pages mentioning "Michael Jordan" are referred to the basketball player, and the total number of pages is also dominating compared to that of any computer scientist.

Our remedy is the hypothesis that although false mentions for an individual entity can be similar to each other, the false mentions across distinct entities belong to more heterogeneous domain than true mentions. Therefore, it is more reliable for a mention to be deemed true if it has similar context with mentions of many *different* entities than with many mentions of *the same* entity name. Our solution is to limit the propagation in an appropriate extent via i) unlinking – disallow the propagation between candidate mentions of the same entity and ii) normalization – restrict the total contribution from mentions of an individual entity.

Along with the normalization, another issue is how to smooth the raw similarity-based weight. Since we select only a short text window for context similarity computation, the similarity score between many pairs of candidate mentions could be zero or close to zero. We smooth the propagation weight by adding a smoothing term.

Formally, we define:

$$\begin{aligned} w_{i'j',ij} &= \frac{z_{ij}}{k}, \text{ if } i = i' &(3)\\ &= \frac{\mu_{i'j',ij}}{V_i Z} + \frac{z_{ij}}{k}, \text{ otherwise} &(4)\\ z_{ij} &= 1 - \frac{\sum_{i' \neq i} \sum_{j'} \mu_{i'j',ij}}{V_i Z} &(5)\\ Z &= \max_{i,j} \frac{\sum_{i' \neq i} \sum_{j'} \mu_{i'j',ij}}{V_i} &(6) \end{aligned}$$

where V_i is the number of documents that have candidate mentions of e_i in the collection, and k is the total number of candidate mentions: $V_i = |\{d_j | (e_i, d_j) \in R\}|, k = |R|$.

Each of the new weight, which is asymmetric, consists of two parts, the normalized context similarity and the smoothing term. For two candidate mentions of the same entity, the first part is zero, as shown in Equation 3. With the denominator V_i in Equation 4, we confine the total score that propagates out of candidate mentions of any single entity.

z_{ij} and Z are constants used for smoothing. z_{ij} controls the weight of smoothing term $\frac{1}{k}$. It is negatively correlated with the overall context similarity of (e_i, d_j) and other mentions. Z is a constant that represents the maximum overall context similarity of one mention with other mentions. If the overall context similarity of one mention with other mentions is high (close to Z), the smoothing term should be small in order to avoid deviating the final weight from the similarity score significantly. Equation 5 and 6 ensures $z_{ij} \geq 0$.

3.3 Solution

We can prove that MentionRank can be rewritten as $\mathbf{r} = \mathbf{M}\mathbf{r}$ where $\mathbf{r}$ is the ranking score vector and $\mathbf{M}$ is a Markov matrix that is *stochastic*, *irreducible*, and *aperiodic*. Therefore, a power method is guaranteed to converge akin PageRank [4, 1].

MentionRank is different from standard PageRank and its variation on weighted and undirected graph in many aspects. First of all, the mentions can be grouped by the entities corresponding to the mention, *e.g.*, the two mentions of Microsoft shaded in Figure 4(a). It has a series of consequences in the modeling.

- Unlinking: PageRank uses links of two vertices to propagate the ranking score; in MentionRank, the score is propagated between candidate mentions for different entities, while the mentions for the same entity are "unlinked" in the mention graph as shown in Figure 4(a).
- Normalization: Regarding the normalization for the propagation weight from a source vertex to a target vertex, PageRank and its variation uses the (weighted) out-degree of the source vertex as the denominator; in MentionRank, we normalize the weight according to i) the number of mentioning documents of the source entity, or *entity degree*; and ii) the maximum overall context similarity of one mention with other mentions.
- Smoothing: PageRank adds random jump $\frac{1}{k}$ to those nodes without any outgoing edges; MentionRank decays the smoothing term of every mention with regard to its overall context similarity with other mentions.

The solution to Equation 1 is a vector with length k, composed of the ranking scores r_{ij}. The scores are globally comparable across different entities. We use an example to show that MentionRank produces better results than standard PageRank.

Example 2. *We alter the example in Figure 2 a little. Assume (e_1, d_1) and (e_2, d_3) are false mentions and the other 2 mentions in d_2 are true; the context similarity between true mentions are 0.8 and all others 0.2. When $\lambda = 0$ (no entity co-mention prior is used), Equation 1 becomes:*

$$\begin{pmatrix} r_{11} \\ r_{12} \\ r_{22} \\ r_{23} \end{pmatrix} = \begin{pmatrix} 0.15 & 0 & 0.2 & 0.35 \\ 0.15 & 0 & 0.8 & 0.35 \\ 0.35 & 0.8 & 0 & 0.15 \\ 0.35 & 0.2 & 0 & 0.15 \end{pmatrix} \begin{pmatrix} r_{11} \\ r_{12} \\ r_{22} \\ r_{23} \end{pmatrix}$$

The solution is $(r_{11}, r_{12}, r_{22}, r_{23}) = (0.4, 1.0, 1.0, 0.4)$. Here we have normalized the scores so that the largest score is 1. The PageRank solution with the same setting is $(0.5, 1.0, 1.0, 0.5)$. *True mentions are better separated from false mentions in MentionRank, although the ranking order is the same. Furthermore, if we change the three unnormalized weight $\mu_{ij,11}$ from 0.2 to 0.5 and maintain all the other weight, PageRank retains the same solution because the first column remains the same after its normalization; MentionRank can respond to the fact that r_{22} and r_{23} should get more credits propagated from r_{11}, and produce higher score for them.*

4. LEVERAGING ADDITIONAL KNOWLEDGE

The basic MentionRank model does not require any additional knowledge about the entities beyond their names. In this section, we discuss how to leverage three sorts of prior knowledge: entity similarity, entity attributes and reference pages.

4.1 Entity Similarity

In TED problem, the target entities are homogeneous. However, the degree of homogeneity among subsets of them could vary slightly. Intuitively, if two entities belong to the same subcategory or have similar attributes, they are more similar to each other than other pairs of target entities. The degree of ranking score interdependency of two candidate mentions should be lower if the corresponding target entities are less similar. Hence, if a user has additional knowledge about the entity similarity, *e.g.*, based on their categorical or numerical attributes, we can use the product of entity similarity and context similarity as the unnormalized link weight instead of context similarity alone.

4.2 Entity Attributes and Reference Pages

In certain cases, users can find representative pages for the target entities from an external knowledge base like Wikipedia. In some other cases, target entities are records stored in a database and have a rich set of attributes. For example, books have authors, publishers and number of pages as their attributes. When the attributes are textual, the occurrence of these attribute names or values in a document near a mention indicates reference to a target entity. Therefore, one can concatenate these attribute names and values to create a pseudo "representative document" [6]. We refer to both the reference pages from the knowledge base and the pseudo representative documents as *authentic documents*. The TED problem can be extended to allow an authentic document set $\{a_i | 1 \leq i \leq n\}$ as additional input,

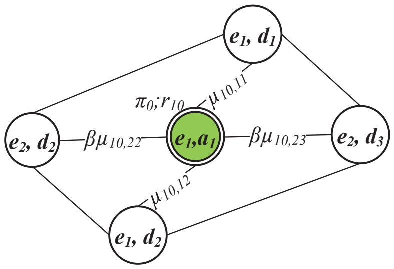


Figure 5: MentionRank with virtual node. Only newly introduced variables are labeled on the graph.

where a_i is the authentic document for entity e_i. For simplicity, we assume one authentic document per entity; our solution, as we will present below, can be easily extended to support multiple authentic documents per entity.

In this extended setting, we can rank candidate mentions using the ContentMatch approaches [8, 6]. However, it has the following issues.

Incomplete set of authentic documents. It is often the case that only a small fraction of entities have authentic documents. In this case, ContentMatch approaches cannot disambiguate mentions of other entities. For those mentions, one rescue solution is to compute the average context similarity with the given authentic documents as the ranking score.

Varying domain-representativeness. Some authentic documents are representative of all the input entities in the domain while some are representative for only the corresponding entity. The above rescue solution does not work well unless the authentic documents are highly domain representative.

To address these issues, we extend our MentionRank model. The authentic documents provide an opportunity for leveraging the interdependency hypothesis, as they can be regarded as confident true mentions. Therefore, we add a *virtual node* in the mention graph for every authentic document and assign a very high prior π_0. Figure 5 shows the new mention graph extended from Figure 4(a) when the authentic document a_1 for e_1 is provided, and the virtual node is marked with double rings. Though we do not need to actually rank the authentic document mention (e_1, a_1), we maintain the score r_{10} so as to propagate evidence from this confident true mention to unknown candidate mentions. We link the virtual node with all the actual candidate mentions. For candidate mentions of the entity corresponding to the virtual node, the link weight is set to the context similarity. For other entities, we introduce a decay factor $\beta \in [0, 1]$ to capture the domain representativeness, and define the link weight as the product of context similarity and β.

This extension resolves both issues we discussed above. First, the candidate mentions of the entity corresponding to the virtual node having high context similarity with the authentic document will receive high credits. Then, these mentions will propagate the score further to similar mentions through the network, including to those entities without authentic documents. Second, our model can adapt to varying domain representativeness of authentic documents. β should be set high only when authentic documents are representative across entities; in such cases, the authentic documents will propagate scores directly to mentions of all entities.

The extended model is consistent with the original MentionRank, and we name it MentionRank+VirtualNode. The same iterative algorithm applies. Eventually, the ranking score of all the mentions is determined by the interaction of three parts — context similarity to authentic documents, context similarity with other mentions and entity co-mention degree.

5. EXPERIMENTAL EVALUATION

We conduct a series of experiments to evaluate the effectiveness of our method.

5.1 Implementation

Our approach has two phases.

Mention graph building. We build the mention graph from the input entity list, documents and candidate mentions. We compute the context similarity, prior estimation and propagation weight. We add virtual nodes and links to the graph if additional authentic documents are given.

MentionRank computation. We implement the power iteration method [1], starting with an initial score 1.0 for every r_{ij}, and apply Equation 1 iteratively until the score change is smaller than a threshold ϵ.

Both steps can be expensive when we have many candidate mentions. We can leverage distributed computational framework for both steps to scale to large datasets. In this paper, we focus on quality and consider single machine implementation.

We use the following settings for MentionRank and MentionRank+VirtualNode unless otherwise specified: $\lambda = 0.5$, $p_{ij} = |\{x|(e_x, d_j) \in R\}|, \pi_0 = 1000, \beta = 0$. During the tf-idf similarity computation, we discard terms with frequency lower than 10, or with document frequency higher than 0.8.

5.2 Datasets

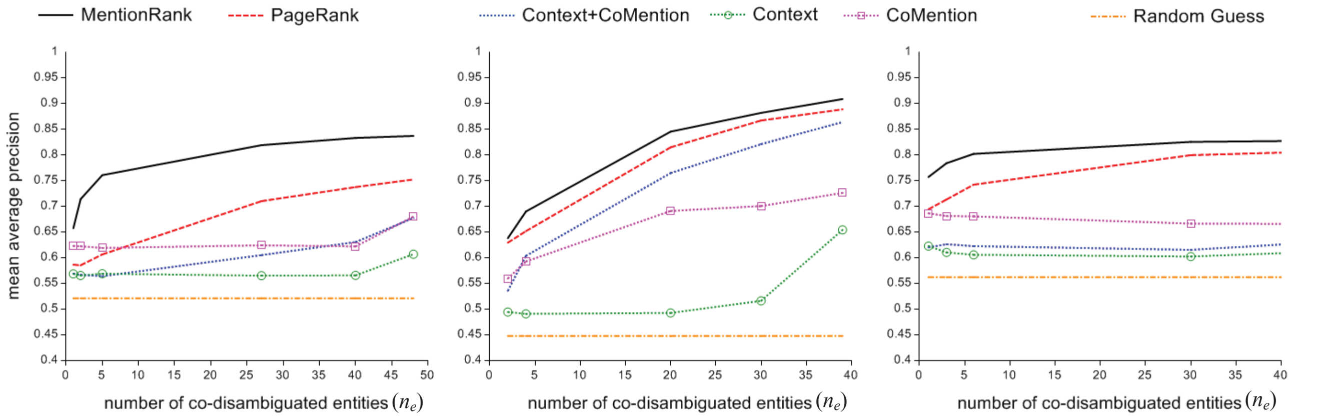
There is no benchmark dataset for our problem: the targeted disambiguation task with ad-hoc homogeneous sets of named entities. We thus create three datasets in 3 different target domains for evaluation: Programming Languages, Science Fiction Books and Sloan Fellows. We chose these domains because (i) many applications require entity extraction in these domains and (ii) there is ambiguity in the entity names. For each domain, we start by obtaining the ad-hoc list of entity names.[2]

• Programming Languages (PL). We collect the language names from two subcategories of *Objected-oriented programming* category of Wikipedia, namely *Class-based programming languages* and *Prototype-based programming*. The former contains 31 languages including Java, Python and Ruby. The latter contains 23 languages including Perl and R. In total there are 54 entities.

• Science Fiction Books (Book). We collect the book names from the first page of 4 subcategories from an online Science Fiction Book Club [3]: Aliens, Alternate History, Military, and Near Future. Each contains 10 book names and in total 40. Some examples are *A Pleasure to Burn*, *Golden Reflections* and *Pathfinder*.

[2] Often, enterprise provides a small set of entities (50 or 100). For example, a shoe company might be interested in mining the sentiment of its own brand and a few competing brands. Hence, performing well for entity sets of such sizes is critical. We plan to experiment with a larger number of entities in future work.

[3] http://www.sfbc.com/science-fiction-books/



(a) PL (b) Book (c) Sloan

Figure 6: Scenario 1 – only entity names are provided.

Table 1: Summary of evaluation datasets

	n	k	#Edge	#Comention	%Positive
PL	54	3,543	1.84M	4,873	43.5
Book	40	3,614	2.06M	978	47.5
Sloan	60	3,651	2.04M	77	82.5

n=# Entity, k=# Mention, #Edge - in mention graph

• Sloan Fellowships (Sloan). We randomly select 60 Sloan fellows in 6 fields in 2010, 10 from each field. The list includes computer scientists Jonathan Kelner, Ben Taskar and Luis Ceze.

After we obtained the entity list, the candidate mention identification component identifies exact occurrences of the entity names in Web pages. From Web pages with candidate mentions, we sample 1% for first two domains and 10% for the third domain. For each dataset we manually label more than 3500 mentions, 50-100 for each entity. These sampled web pages are of a variety of types: commercial sites, news articles, blogs, forum posts and so on.

Naturally, the hardness of disambiguation on top of these datasets is different due to the different extent of ambiguity, page sanity and domain specialty. For example, the fraction of true mentions in all the candidate mentions is one characterization of the ambiguity; the expected performance of random guess is another. We summarize the characteristics of the three datasets in Table 1. Sloan dataset has the largest fraction of positive labels, but fewest co-mentions. The other two datasets are similar in terms of ambiguity, with no more than half of the candidate mentions inside the target domain. Surprisingly, the density of the mention graph is similar across the three domains.

5.3 Experimental Results

5.3.1 *Scenario 1 – Entity Name Only*

In the most challenging setting, only entity names are provided as input. None of the ContentMatch methods relying on external knowledge bases can be applied. We evaluate the performance of MentionRank in this challenging case, and decompose its elements to see their relative importance.

• *Context* is a simple solution relying on context similarity only. It computes the average context similarity of each mention with all the other mentions and rank them accordingly.

• *CoMention* only relies on the co-mention prior to rank all the mentions.

• *Context+CoMention* uses a linear combination of them.

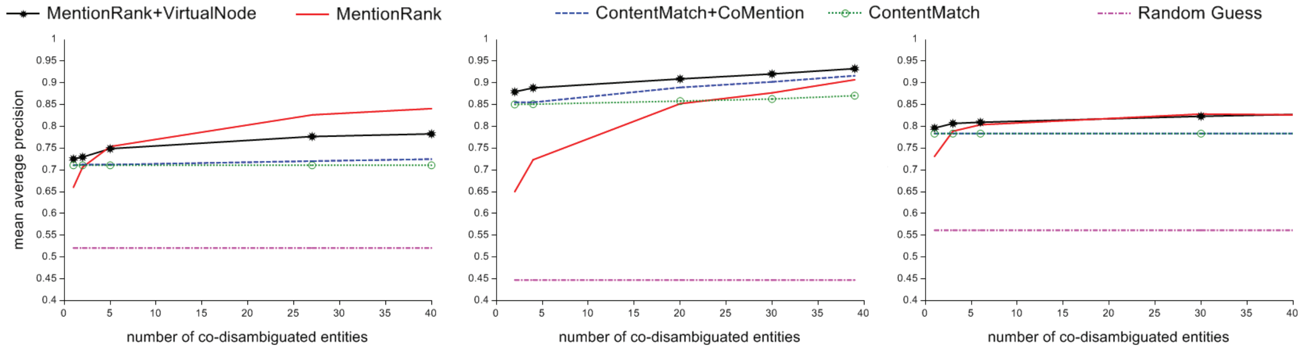
• *PageRank* refers to the direct application of PageRank on the weighted undirected mention graph.

MentionRank, PageRank and Context+CoMention capture both co-mention and context similarity. While MentionRank and PageRank capture the interdependency by propagating the ranking scores (albeit in different ways), Context+CoMention does not perform the propagation and does not capture the interdependency.

Different users have different preference to precision and recall, so we evaluate the performance in average cases with the mean average precision (MAP) measure. Since we perform collective disambiguation, the performance is dependent on the size of the collection. Hence, we fix the number n_e of entities each entity is co-disambiguated with and measure MAP. We vary n_e and plot the MAP for each value of n_e in Figure 6.

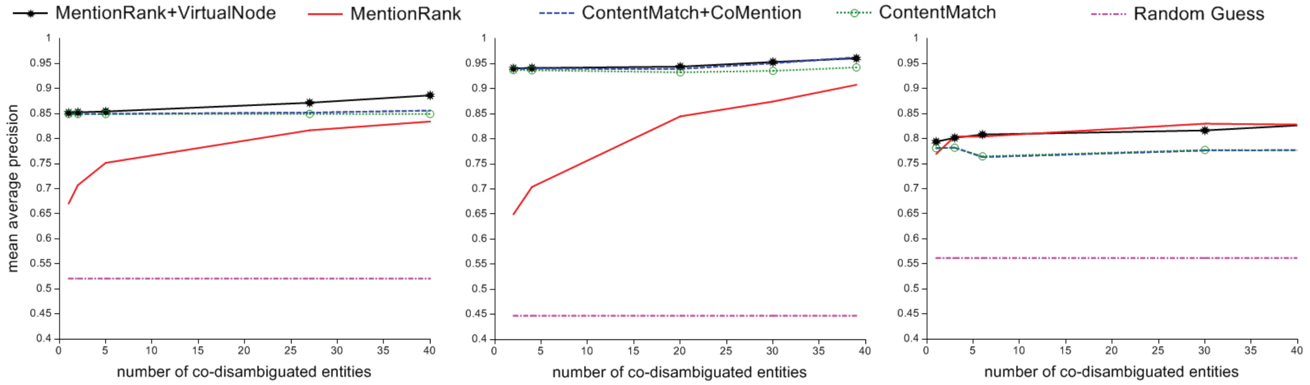
The performance of different algorithms all grow when more entities are co-disambiguated, but MentionRank significantly outperforms other methods. We achieve more than 80% MAP in all three datasets when 20 or more entities are provided. Of the two simplest solution with a single component, CoMention has in general better performance than Context. Neither can go over 70% MAP in Book while MentionRank reaches as high as 90%. Embracing both components is a more promising strategy. However, the simple combination of them Context+CoMention does not give satisfactory results. The two graph-based models have ranking results of much better quality, of which MentionRank outperforms PageRank in most cases, with largest margin in PL dataset (25%). As one particular example, the book *How Firm a Foundation* has only one true mention in the 56 sampled pages. MentionRank ranks it higher than all the remaining 55 false mentions; PageRank ranks it 2nd place, after a mention referring to a Mormon song; Context ranks it after 6 false mentions about the song.

To summarize, MentionRank leverages all the 3 hypotheses, namely context similarity, co-mention and interdependency and removal of any one will degrade the performance. It is also a better model compared with standard PageRank.



(a) PL (b) Book (c) Sloan

Figure 7: Scenario 2 – entity names and attributes are provided.



(a) PL (b) Book (c) Sloan

Figure 8: Scenario 3 – entity names and authentic pages are provided.

5.3.2 *Scenario 2 - Entity Name + Attributes*

We extract the following attributes for the three datasets we created.

- PL (from Wikipedia). Paradigm; Developed by; Typing discipline; Platform.
- Book (from Amazon catalog). Author; Hardcover/ paperback; #Pages; Publisher; Language; ISBN; Product Dimensions; Shipping Weight.
- Sloan. Affiliation; Position; Research Areas; Email.

The attribute names and values are concatenated into a pseudo representative document for every entity.

For this extended TED problem, we compare with the following approaches as baseline:

- ContentMatch [6], using the average occurrence similarity with the pseudo document as the ranking score of every mention;
- ContentMatch +CoMention [14], using a weighted linear combination of the ContentMatch score and the co-mention prior.

We show the same kind of plot as scenario 1 in Figure 7. In all cases, MentionRank + VirtualNode outperforms the two baseline methods. Without using any entity attributes, MentionRank achieves comparable quality as methods relying on attributes, especially when the number of co-disambiguated entities is large. That validates the power of collective disambiguation in our model.

5.3.3 *Scenario 3 - Entity Name + Authentic Pages*

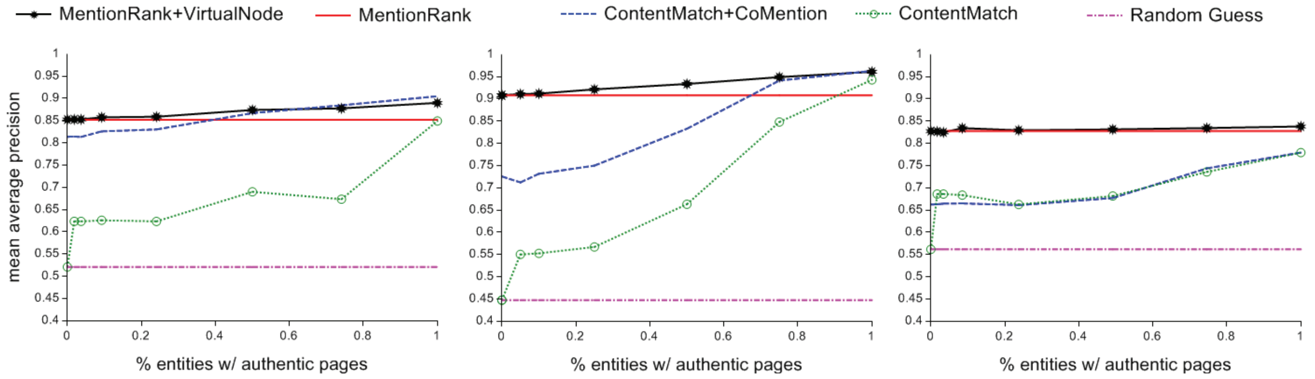
This scenario is akin scenario 2, except that we now have real authentic pages than pseudo ones comprised of entity attributes. We use Wikipedia pages for PL dataset, product pages on sfbc.com for Book, and researcher homepages for Sloan. We made the baseline stronger by removing the entity name itself from the context of a mention before the computation of the ContentMatch score. That reduces the spurious high context similarities caused by many occurrences of the entity name in the context of a false mention.

MentionRank with virtual node still performs the best, with 89%, 96% and 85% MAP on the three datasets. Comparing Figure 8 and Figure 7, we find that the quality of the authentic pages is higher than that of the attributes, and boost the performance of ContentMatch-based methods in the first two datasets. For Sloan dataset, the homepages are not very representative and do not help much.

5.3.4 *Scenario 4 - Entity Name + Incomplete Set of Authentic Pages*

A user may be able to obtain authentic pages for a small fraction of entities (say, the ones present in a knowledge base) but not all the entities. We evaluate the performance of MentionRank+VirtualNode in dealing with the incomplete set of authentic pages. The disambiguation is performed for all input entities together, and the fraction of entities with authentic pages is varied.

As shown in Figure 9, our method without parameter tuning outperforms baseline with a significant margin in most cases, especially when the fraction of entities with authentic pages is small. When only 2-5% such pages are available,



(a) PL (b) Book (c) Sloan

Figure 9: Scenario 4 – entity names and part of authentic pages are provided.

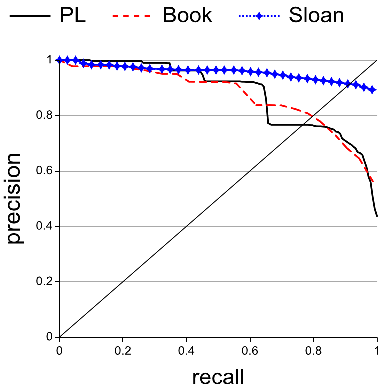


Figure 10: Precision-recall curve for MentionRank with entity name as input.

the MAP margin with ContentMatch + CoMention is more than 25% for Book and Sloan datasets.

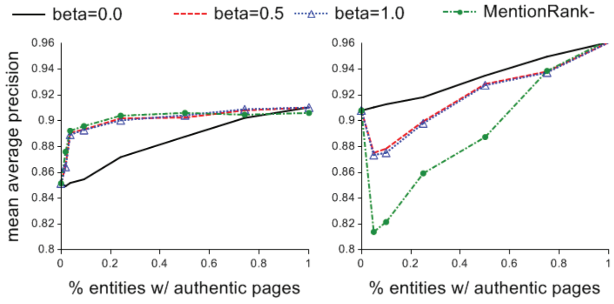
Based on Figure 9 and Figure 6, we find that MentionRank works considerably well when none of the authentic pages are provided, compared with the case even when a complete set of authentic pages are given. The difference of MAP in those two extreme cases is smaller than 5% in the three test datasets, implying our method can save significant amount of user effort without losing much accuracy in practice.

5.3.5 Thresholding

We discuss how to retrieve relevant mentions from the ranking results to trade off precision and recall. In general one has the following strategies:

- retrieve top-t or top-f% (e_i, d_j) mentions;
- retrieve mentions with ranking score larger than t, while the score is normalized such that the largest score is 1;
- retrieve top mentions with the sum of ranking scores larger than t, while the score is normalized such that the sum of all scores is 1.

For each strategy, one can refer to mixed ranking for all mentions or per-entity ranking, depending on the application. We show the precision-recall curve for the mixed ranking (for MentionRank with entity name only) in Figure 10 to give an idea what precision and recall to expect with different thresholding. We first observe that the precision remains over 0.95 (resp. 0.9) in order to obtain a recall of 0.4 (resp. 0.6). That implies *if extracting half of the true mentions is sufficient for the application, a very high precision can be obtained.* Next, we check the *break-even* point where precision is equal to recall, and find it is around 0.8 in PL and Book, and around 0.9 in Sloan. Referring back to the positive ratio of each dataset, we know that when retrieving top 40-50% mentions from PL or Book, one expects 0.8 for both precision and recall; when retrieving top 80% mentions from Sloan, one expects 0.9 for both precision and recall.



(a) PL (b) Book

Figure 11: Examples of β's effect.

5.3.6 Parameter Selection

For selection of the three parameters $\lambda \in [0,1], \pi_0 \in [0,+\infty)$, and $\beta \in [0,1]$, we discuss their effect one by one. λ is the only parameter in MentionRank without virtual node, and controls the weight between prior and propagated score from other mentions. MentionRank is insensitive to this parameter when λ is varied from 0.2 to 0.8. We do not show the curves as they overlapped with each other.

π_0 determines overall how much we trust the authentic documents; it should be set according to the confidence on the authentic document quality. For example, for PL when the Wikipedia pages are available, $\pi_0 \in [1K, 10K]$ is good; while for Sloan where author pages are used as authentic documents, $\pi_0 \in [10, 100]$ is better.

β only matters when authentic documents for every entity are not available, and should be set depending on the domain representativeness of authentic documents. Figure 11 shows the effect of β, and the comparison with a simpler solution MentionRank–, for which we do not use virtual node but simply change the prior of a mention in MentionRank into its ContentMatch+CoMention score (average similarity with all authentic documents is used if its own authentic document is unavailable). When authentic documents are domain representative, like in PL, higher β is preferred; when each of them has high quality but is not representative across entities, like in Book, lower β is preferred; when they have low quality, like in Sloan, the performance is insensitive to β as π_0 is small. The simpler solution MentionRank– works well only in the first case (where domain representativeness is high), and MentionRank+VirtualNode outperforms it by around 10% otherwise.

6. RELATED WORK

Named entity disambiguation has been extensively studied in the literature [5, 9, 17, 18, 8, 15, 14, 12, 13]. The prior work can be classified into two broad categories:

- *Independent mention disambiguation* where each mention is mapped to a knowledge base entity independently. The main idea is to compare the context of the mention with the text metadata associated with the entity in the knowledge base [5, 9, 17, 8]. They differ in the features used (e.g., bag of words vs. Wikipedia categories) and the comparison technique (cosine similarity vs. classifier). The main drawback is that they do not consider interdependence between disambiguation decisions.
- *Intra-document collective mention disambiguation* which observe that a document typically refers to topically coherent entities. They consider interdependence between disambiguation decisions *within a document* [18, 15, 14, 12]. They perform *collective assignment* of candidate mentions in a document to entities and selects an assignment that not only maximizes the mention context-to-entity similarity but also the coherence among the assigned entities [15, 14]. Coherence between a pair of entities is computed using the knowledge base, e.g., based on the number of common Wikipedia pages that link to Wiki pages of these two entities [15]. While some approaches model the interdependence as sum of their pair-wise dependencies [18, 15], more recent techniques model the global interdependence [14, 12].

These studies consider only entities present in a knowledge base; we focus on *ad-hoc* entities. The EROCS system identifies mentions of entities in an enterprise database from text documents; it relies on a rich set of attributes in the database and disambiguates based on the similarity between the attribute values and the context of the mention [6]. However, it does not consider coherence among entities.

To the opposite of targeted entity disambiguation, the untargeted disambiguation problem aims to partition the mentions of different entities with the same name. Researchers solve it by clustering the candidate mentions such that the mentions in a cluster refer to the same entity [16, 19]. The main idea is to extract features from the context of each mention and cluster them based on those features (e.g., using agglomerative clustering or graph partitioning). The solution cannot be directly applied to TED.

Entity resolution is related to the problem of named entity disambiguation: the goal is to identify the entities referenced in database records as opposed to text documents. Recently collective algorithms to exploit interdependencies among references in this setting have been proposed [3].

7. CONCLUSIONS

In this paper, we study the novel problem of targeted disambiguation of ad-hoc, homogeneous sets of entities. We develop a novel graph-based model MentionRank to address the challenge posed by adhoc-ness via leveraging the homogeneity constraint. Our experiments show that without any additional knowledge about the entities, our model achieves similar quality as previous approaches that rely on additional knowledge; with additional knowledge incorporated, it outperforms those previous approaches.

Our work can be extended in multiple directions. In terms of quality, it is worth exploring more advanced features for measuring similarity between contexts (*e.g.*, according to the nature of the documents). One can refine the co-mention prior by considering the distance of co-mentions and capture the entity homogeneity more precisely with the help of ontology. In terms of scalability, it is promising to explore approximation of our model for speedup, *e.g.*, with graph sparsification or clustering techniques.

8. REFERENCES

[1] A. Arasu, J. Novak, A. Tomkins, and J. Tomlin. PageRank computation and the structure of the web: Experiments and algorithms. In *WWW*, 2002.

[2] J. Artiles et al. WePS-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In *Proc. Conf. Multilingual and Multimodal Information Access Evaluation (CLEF)*, 2010.

[3] I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1):1–36, 2007.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[5] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, 2006.

[6] V. T. Chakaravarthy, H. Gupta, P. Roy, and M. Mohania. Efficiently linking text documents with relevant structured information. In *VLDB*, 2006.

[7] T. Cheng, H. W. Lauw, and S. Paparizos. Entity synonyms for structured web search. *IEEE Trans. Knowl. Data Eng.*, PP(99):1, 2011.

[8] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, 2007.

[9] S. Dill et al. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *WWW*, 2003.

[10] W. A. Gale, K. W. Church, and D. Yarowsky. One sense per discourse. In *Proc. Workshop on Speech and Natural Language*, 1992.

[11] C. H. Gooi and J. Allan. Cross-Document coreference on a large scale corpus. In *HLT-NAACL*, 2004.

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

[13] X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *CIKM*, 2009.

[14] J. Hoffart et al. Robust disambiguation of named entities in text. In *EMNLP*, 2011.

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

[16] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *CONLL*, 2003.

[17] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, 2007.

[18] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, 2008.

[19] L. Sarmento, A. Kehlenbeck, E. Oliveira, and L. Ungar. An approach to web-scale named-entity disambiguation. In *Proc. Intl. Conf. Machine Learning and Data Mining in Pattern Recognition (MLDM)*, 2009.