Linking Entities to a Knowledge Base with Query Expansion

Swapna Gottipati
School of Information Systems
Singapore Management University
Singapore
swapnag.2010@smu.edu.sg

Jing Jiang
School of Information Systems
Singapore Management University
Singapore
jingjiang@smu.edu.sg

Abstract

In this paper we present a novel approach to entity linking based on a statistical language model-based information retrieval with query expansion. We use both local contexts and global world knowledge to expand query language models. We place a strong emphasis on named entities in the local contexts and explore a positional language model to weigh them differently based on their distances to the query. Our experiments on the TAC-KBP 2010 data show that incorporating such contextual information indeed aids in disambiguating the named entities and consistently improves the entity linking performance. Compared with the official results from KBP 2010 participants, our system shows competitive performance.

1 Introduction

When people read news articles, Web pages and other documents online, they may encounter named entities which they are not familiar with and therefore would like to look them up in an encyclopedia. It would be very useful if these entities could be automatically linked to their corresponding encyclopedic entries. This task of linking mentions of entities within specific contexts to their corresponding entries in an existing knowledge base is called *entity linking* and has been proposed and studied in the Knowledge Base Population (KBP) track of the Text Analysis Conference (TAC) (McNamee and Dang, 2009). Besides improving an online surfer's browsing experience, entity linking also has potential usage in many other applications such as normalizing entity mentions for information extraction.

The major challenge of entity linking is to resolve name ambiguities. There are generally two types of ambiguities: (1) Polysemy: This type of ambiguities refers to the case when more than one entity shares the same name. E.g. *George Bush* may refer to the 41st President of the U.S., the 43rd President of the U.S., or any other individual who has the same name. Clearly polysemous names cause difficulties for entity linking. (2) Synonymy: This type of ambiguities refers to the case when more than one name variation refers to the same entity. E.g. *Metro-Goldwyn-Mayer Inc.* is often abbreviated as *MGM*. Synonymy affects entity linking when the entity mention in the document uses a name variation not covered in the entity's knowledge base entry.

Intuitively, to disambiguate a polysemous entity name, we should make use of the context in which the name occurs, and to address synonymy, external world knowledge is usually needed to expand acronyms or find other name variations. Indeed both strategies have been explored in existing literature (Zhang et al., 2010; Dredze et al., 2010; Zheng et al., 2010). However, most existing work uses supervised learning approaches that require careful feature engineering and a large amount of training data. In this paper, we take a simpler unsupervised approach using statistical language model-based information retrieval. We use the KL-divergence retrieval model (Zhai and Lafferty, 2001) and expand the query language models by considering both the local contexts within the query documents and global world knowledge obtained from the Web.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 804–813,
Edinburgh, Scotland, UK, July 27–31, 2011.

Symbol	Description
Q	Query
D_Q	Query document
N_Q	Query name string
E	KB entity node
N_E	KB entity name string
D_E	KB entity disambiguation text
$\mathcal{S}_Q$	Set of alternate query name strings
$N_Q^{l,i}$	Local alternative name strings
N_Q^g	Global alternative name strings
$\mathcal{E}_Q$	Candidate KB entries for Q
θ_Q	Query Language Model
θ_Q^L	KB entry language model using local context from D_Q
θ_Q^G	KB entry language model using global knowledge
θ_Q^{L+G}	KB entry language model using local context and global knowledge
θ_{N_E}	KB entry language model with named entities only
$\theta_{N_E+D_E}$	KB entry language model with named entities and disambiguation text

Table 1: Notation

We evaluate our retrieval method with query expansion on the 2010 TAC-KBP data set. We find that our expanded query language models can indeed improve the performance significantly, demonstrating the effectiveness of our principled and yet simple techniques. Comparison with the official results from KBP participants also shows that our system is competitive. In particular, when no disambiguation text from the knowledge base is used, our system can achieve an overall 85.2% accuracy and 9.3% relative improvement over the best performance reported in KBP 2010.

2 Task Definition and System Overview

Following TAC-KBP (Ji et al., 2010), we define the entity linking task as follows. First, we assume the existence of a Knowledge Base (KB) of entities. Each KB entry E represents a unique entity and has three fields: (1) a name string N_E, which can be regarded as the official name of the entity, (2) an entity type T_E, which is one of {PER, ORG, GPE, UNKNOWN}, and (3) some disambiguation text D_E. Given a query Q which consists of a query name string N_Q and a query document D_Q where the name occurs, the task is to return a single KB entry to which the query name string refers or *Nil* if there is no such KB entry.

It is fairly natural to address entity linking by ranking the KB entries given a query. In this section we present an overview of our system, which consists of two major stages: a candidate selection stage to identify a set of candidate KB entries through name matching, and a ranking stage to link the query entity to the most likely KB entry. In both stages, we consider the query's local context in the query document and world knowledge obtained from the Web. It is important to note that the selection stage is based on string matching where the order of the word matters. It is different from the ranking stage where a probabilistic retrieval model based on bag-of-word representation is used. Our preliminary experiments demonstrate that without the first candidate selection stage the linking process results in low performance.

2.1 Selecting Candidate KB Entries

The first stage of our system aims to filter out irrelevant KB entries and select only a set of candidates that are potentially the correct match to the query. Intuitively, we determine whether two entities are the same by comparing their name strings. We therefore need to compare the query name string N_Q with the name string N_E of each KB entry. However, because of the name ambiguity problem, we cannot expect the correct KB entry to always have exactly the same name string as the query. To address this problem, we use a set of alternative name strings expanded from N_Q and select KB entries whose name

strings match at least one of them. These alternative name strings come from two sources: the query document D_Q and the Web.

First, we observe that some useful alternative name strings come from the query document. For example, a PER query name string may contain only a person's last name but the query document contains the person's full name, which is clearly a less ambiguous name string to use. Similarly, a GPE query name string may contain only the name of a city or town but the query document contains the state or province, which also helps disambiguate the query entity. Based on this observation, we do the following. Given query Q, let $\mathcal{S}_Q$ denote the set of alternative query name strings. Initially $\mathcal{S}_Q$ contains only N_Q. We then use an off-the-shelf NER tagger to identify named entities from the query document D_Q. For PER and ORG queries, we select named entities in D_Q that contain N_Q as a substring. For GPE queries, we select named entities that are of the type GPE, and we then combine each of them with N_Q. We denote these alternative name strings as $\{N_Q^{l,i}\}_{i=1}^{K_Q}$, where l indicates that these name strings come locally from D_Q and K_Q is the total number of such name strings. $\{N_Q^{l,i}\}$ are added to $\mathcal{S}_Q$. Figure 1 and Figure 2 show two example queries together with their $\mathcal{S}_Q$.

Sometimes alternative name strings have to come from external knowledge. For example, one of the queries we have contains the name string "AMPAS," and the query document also uses only this acronym to refer to this entity. But the full name of the entity, "Academy of Motion Pictures Arts and Sciences," is needed in order to locate the correct KB entry. To tackle this problem, we leverage Wikipedia to find the most likely official name. Given query name string N_Q, we check whether the following link exists: `http://en.wikipedia.org/`N_Q. If N_Q is an abbreviation, Wikipedia will redirect the link to the Wikipedia page of the corresponding entity with its official name. So if the link exists, we use the title of the Wikipedia page as another alternative name string for N_Q. We refer to this name string as N_Q^g to indicate that it is a global name variant. N_Q^g is also added to $\mathcal{S}_Q$. Figure 2 shows such an example.

For each name string N in $\mathcal{S}_Q$, we find KB entries whose name strings match N. We take the union of these sets of KB entries and refer to it as $\mathcal{E}_Q$. These are the candidate KB entries for query Q.

Query name string (N_Q): *Mobile*
Query document (D_Q): *The site is near Mount Vernon in the Calvert community on the Tombigbee River, some 25 miles (40 kilometers) north of Mobile. It's on a river route to the Gulf of Mexico and near Mobile's rails and interstates. Along with tax breaks and $400 million (euro297 million) in financial incentives, Alabama offered a site with a route to a Brazil plant that will provide slabs for processing in Mobile.*
Alternative Query Strings ($\mathcal{S}_Q$):
from local context: *Mobile, Mobile Mount Vernon, Mobile Calvert, Mobile River, Mobile Mexico, Mobile Alabama, Mobile Brazil*

Figure 1: An example *GPE* query from TAC 2010.

Query name string (N_Q): *Coppola*
Query document (D_Q): *I had no idea of all these semi-obscure connections, felicia! Alex Greenwald and Claire Oswalt aren't names I'm at all familiar with, but Jason Schwartzman I've heard of. Isn't he Sophia Coppola's cousin? I think I once saw a picture of him sometime ago*
Alternative Query Strings ($\mathcal{S}_Q$):
from local context: *Coppola, Sophia Coppola, Sofia Coppola*
from world knowledge(Wikipedia): *Sofia Coppola*

Figure 2: An example *PER* query from TAC 2010.

2.2 Ranking KB Entries

Given the candidate KB entries $\mathcal{E}_Q$, we need to decide which one of them is the correct match. We adopt the widely-used KL-divergence retrieval model, a statistical language model-based retrieval method proposed by Lafferty and Zhai (2001). Given a KB entry E and query Q, we score E based on the KL-divergence defined below:

$$s(E,Q) = -\mathit{Div}(\theta_Q \| \theta_E) = -\sum_{w \in V} p(w|\theta_Q) \log \frac{p(w|\theta_Q)}{p(w|\theta_E)}. \quad (1)$$

Here θ_Q and θ_E are the query language model and the KB entry language model, respectively. A language model here is a multinomial distribution over words (i.e. a unigram language model). V is the vocabulary and w is a single word.

To estimate θ_E, we follow the standard maximum likelihood estimation with Dirichlet smooth-

ing (Zhai and Lafferty, 2004):

$$p(w|\theta_E) = \frac{c(w,E) + \mu p(w|\theta_C)}{|E| + \mu}, \quad (2)$$

where $c(w,E)$ is the count of w in E, $|E|$ is the number of words in E, θ_C is a background language model estimated from the whole KB, and μ is the Dirichlet prior. Recall that E contains N_E, T_E and D_E. We consider using either N_E only or both N_E and D_E to obtain $c(w,E)$ and $|E|$. We refer to the former estimated θ_E as θ_{N_E} and the latter as $\theta_{N_E+D_E}$.

To estimate θ_Q, typically we can use the empirical query word distribution:

$$p(w|\theta_Q) = \frac{c(w,N_Q)}{|N_Q|}, \quad (3)$$

where $c(w,N_Q)$ is the count of w in N_Q and $|N_Q|$ is the length of N_Q. We call this model the *original* query language model.

After ranking the candidate KB entries in $\mathcal{E}_Q$ using Equation (1), we perform entity linking as follows. First, using an NER tagger, we determine the entity type of the query name string N_Q. Let T_Q denote this entity type. We then pick the top-ranked KB entry whose score is higher than a threshold τ and whose T_E is the same as T_Q. The system links the query entity to this KB entry. If no such entry exists, the system returns *Nil*.

3 Query Expansion

We have shown in Section 2.1 that using the original query name string N_Q itself may not be enough to obtain the correct KB entry, and additional words from both the query document and external knowledge can be useful. However, in the KB entry selection stage, these additional words are only used to enlarge the set of candidate KB entries; they have not been used to rank KB entries. In this section, we discuss how to expand the query language model θ_Q with these additional words in a principled way in order to rank KB entries based on how likely they match the query entity.

3.1 Using Local Contexts

Let us look at the example from Figure 2 again. During the KB entry ranking stage, if we use θ_Q estimated from N_Q, which contains only the word "Coppola," the retrieval function is unlikely to rank the correct KB entry on the top. But if we include the contextual word "Sophia" from the query document when estimating the query language model, KL-divergence retrieval model is likely to rank the correct KB entry on the top. This idea of using contextual words to expand the query is very similar to (pseudo) relevance feedback in information retrieval. We can treat the query document D_Q as our only feedback document.

Many different (pseudo) relevance feedback methods have been proposed. Here we apply the relevance model (Lavrenko and Croft, 2001), which has been shown to be effective and robust in a recent comparative study (Lv and Zhai, 2009). We first briefly review the relevance model. Given a set of (pseudo) relevant documents $\mathcal{D}_r$, where for each $D \in \mathcal{D}_r$ there is a document language model θ_D, we can estimate a feedback language model θ_Q^{fb} as follows:

$$p(w|\theta_Q^{\text{fb}}) \propto \sum_{D \in \mathcal{D}_r} p(w|\theta_D) p(\theta_D) p(Q|\theta_D). \quad (4)$$

For our problem, since we have only a single feedback document D_Q, the equation above can be simplified. In fact, in this case the feedback language model is the same as the document language model of the only feedback document, i.e. θ_{D_Q}.

We then linearly interpolate the feedback language model with the original query language model to form an expanded query language model:

$$p(w|\theta_Q^L) = \alpha p(w|\theta_Q) + (1-\alpha) p(w|\theta_{D_Q}), \quad (5)$$

where α is a parameter between 0 and 1, to control the amount of feedback. The larger α is, the less we rely on the local context. L indicates that the query expansion comes from local context. This θ_Q^L can then replace θ_Q in Equation (1) to rank KB entries.

Special Treatment of Named Entities

Usually the document language model θ_{D_Q} is estimated using the entire text from D_Q. For entity linking, we suspect that named entities surrounding the query name string in D_Q are particularly useful for disambiguation and thus should be emphasized over other words. This can be done by weighting

NE and non-NE words differently. In the extreme case, we can use only NEs to estimate the document language model θ_{D_Q} as follows:

$$p(w|\theta_{D_Q}) = \frac{1}{K_Q}\sum_{i=1}^{K_Q}\frac{c(w, N_Q^{l,i})}{|N_Q^{l,i}|}, \quad (6)$$

where $\{N_Q^{l,i}\}$ are defined in Section 2.

Positional Model

Another observation is that words closer to the query name string in the query document are likely to be more important than words farther away. Intuitively, we can use the distance between a word and the query name string to help weigh the word. Here we apply a recently proposed positional pseudo relevance feedback method (Lv and Zhai, 2010). The document language model θ_{D_Q} now has the following form:

$$p(w|\theta_{D_Q}) = \frac{1}{K_Q}\sum_{i=1}^{K_Q} f(p_i, q)\cdot\frac{c(w, N_Q^{l,i})}{|N_Q^{l,i}|}, \quad (7)$$

where p_i and q are the absolute positions of $N_Q^{l,i}$ and N_Q in D_Q. The function f is Gaussian function defined as follows:

$$f(p, q) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\Big(\frac{-(p-q)^2}{2\sigma^2}\Big). \quad (8)$$

where variance σ controls the spread of the curve.

3.2 Using Global World Knowledge

Similar to the way we incorporate words from D_Q into the query language model, we can also construct a feedback language model using the most likely official name of the query entity obtained from Wikipedia. Specifically, we define

$$p(w|\theta_{N_Q^g}) = \frac{c(w, N_Q^g)}{|N_Q^g|}. \quad (9)$$

We can then linearly interpolate $\theta_{N_Q^g}$ with the original query language model θ_Q to form an expanded query language model θ_Q^G:

$$p(w|\theta_Q^G) = \alpha p(w|\theta_Q) + (1-\alpha)p(w|\theta_{N_Q^g}). \quad (10)$$

Here G indicates that the query expansion comes from global world knowledge.

Entity Type	%Nil	%non-Nil
GPE	32.8%	67.2 %
ORG	59.5%	40.5 %
PER	71.7%	28.3 %

Table 2: Percentages of Nil and non-Nil queries.

3.3 Combining Local Context and World Knowledge

We can further combine the two kinds of additional words into the query language model as follows:

$$p(w|\theta_Q^{L+G}) = \alpha p(w|\theta_Q) + (1-\alpha)\Big(\beta p(w|\theta_{D_Q}) + (1-\beta)p(w|\theta_{N_Q^g})\Big). \quad (11)$$

Note that here we have two parameters α and β to control the amount of contributions from the local context and from global world knowledge.

4 Experiments

4.1 Experimental Setup

Data Set: We evaluate our system on the TAC-KBP 2010 data set (Ji et al., 2010). The knowledge base was constructed from Wikipedia with 818,741 entries. The data set contains 2250 queries and query documents come from news wire and Web pages. Around 45% of the queries have non-Nil entries in the KB. Some statistics of the queries are shown in Table 2.

Tools: In our experiments, to extract named entities within D_Q and to determine T_Q, we use the Stanford NER tagger[1]. An example output of the NER tagger is shown below:

```
<PERSON>Hugh Jackman<PERSON> is Jacked!!
```

This piece of text comes from a query document where the query name string is "Jackman." We can see that the NER tagger can help locate the full name of the person.

We use the Lemur/Indri[2] search engine for retrieval. It implements the KL-divergence retrieval model as well as many other useful functionalities.

Evaluation Metric: We adopt the *Micro-averaged accuracy* metric, which is the mean accuracy over all queries. It was used in TAC-KBP 2010 (Ji et

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

[2] http://www.lemurproject.org/indri.php

al., 2010) as the official metric to evaluate the performance of entity linking. This metric is simply defined as the percentage of queries that have been correctly linked.

Methods to Compare: Recall that our system consists of a KB entry selection stage and a KB entry ranking stage. At the selection stage, a set $\mathcal{S}_Q$ of alternative name strings are used to select candidate KB entries. We first define a few settings where different alternative name string sets are used to select candidate KB entries:

- **Q** represents the baseline setting which uses only the original query name string N_Q to select candidate KB entries.
- **Q+L** represents the setting where alternative name strings obtained from the query document D_Q are combined with N_Q to select candidate KB entries.
- **Q+G** represents the setting where the alternative name string obtained from Wikipedia is combined with N_Q to select candidate KB entries.
- **Q+L+G** represents the setting as we described in Section 2.1, that is, alternative name strings from both D_Q and Wikipedia are used together with N_Q to select candidate KB entries.

After selecting candidate KB entries, in the KB entry ranking stage, we have four options for the query language model and two options for the KB entry language model. For the query language model, we have (1) θ_Q, the original query language model, (2) θ_Q^L, an expanded query language model using local context from D_Q, (3) θ_Q^G, an expanded query language model using global world knowledge, and (4) θ_Q^{L+G}, an expanded query language model using both local context and global world knowledge. For the KB entry language model, we can choose whether or not to use the KB disambiguation text D_E and obtain θ_{N_E} and $\theta_{N_E+D_E}$, respectively.

4.2 Results and Discussion

First, we compare the performance of KB entry selection stage for all four settings on non-Nil queries. The performance measure recall is defined as

$$recall = \begin{cases} 1, & if\ E\ that\ refers\ to\ Q,\ exists\ in\ \mathcal{E}_Q \\ 0, & otherwise \end{cases}$$

The recall statistics in Table 3 shows that, Q+L+G has the highest recall of the KB candidate entries.

Method	Recall(%)
Q	67.1
Q+L	89.7
Q+G	94.9
Q+L+G	98.2

Table 3: Comparing the effect of candidate entry selection using different methods - KB entry selection stage recall.

Before examining the effect of query expansion in ranking, we now compare the effect of using different sets of alternative query name strings in the candidate KB entry selection stage. For this set of experiments, we fix the query language model to θ_Q and the KB entry language model to θ_{N_E} in the ranking stage.

Table 4 shows the performance of all the settings in terms of micro-averaged accuracy. The results shown in Tables 4, 5 and 6 are based on the optimum parameter settings. We can see that in terms of the overall performance, both Q+L and Q+G give better performance than Q with a 7.7% and a 9.9% relative improvement, respectively. Q+L+G gives the best performance with a 12.8% relative improvement over Q. If we further zoom into the results, we see that for ORG and PER queries, when no correct KB entry exists (i.e. the Nil case), the performance of Q, Q+L, Q+G and Q+L+G is very close, indicating that the additional alternative query name strings do not help. It shows that the alternative query name strings are most useful for queries that do have their correct entries in the KB.

We now further analyze the impact of the expanded query language models θ_Q^L, θ_Q^G and θ_Q^{L+G}. We first analyze the results without using the KB disambiguation text, i.e. using θ_{N_E}. Table 5 shows the comparison between θ_Q and other expanded query language models in terms of micro-averaged accuracy. The results reveal that the expanded query language models can indeed improve the overall performance (the both Nil and non-Nil case) under all settings. This shows the effectiveness of using the principled query expansion technique coupled with KL-divergence retrieval model to rank KB entries.

Method	All				Nil			Non-Nil		
	ALL	GPE	ORG	PER	GPE	ORG	PER	GPE	ORG	PER
Q	0.6916	0.5714	0.6533	0.8495	0.8618	**0.9888**	**0.9963**	0.4294	0.1612	0.4789
Q+L	0.7449	0.7156	0.6533	0.8655	**0.9472**	**0.9888**	0.9944	0.6024	0.1612	0.5399
Q+G	0.7604	0.7009	**0.6893**	0.8908	0.9431	**0.9888**	0.9944	0.5825	**0.2500**	0.6291
Q+L+G	**0.7800**	**0.7583**	**0.6893**	**0.8921**	0.9431	**0.9888**	0.9944	**0.6680**	**0.2500**	**0.6338**

Table 4: Comparing the performance of using different sets of query name strings for candidate KB entry selection. θ_Q and θ_{N_E} are used in KB entry ranking.

Method	QueryModel	All				Nil			Non-Nil		
		ALL	GPE	ORG	PER	GPE	ORG	PER	GPE	ORG	PER
Q+L	θ_Q	0.7449	0.7156	**0.6533**	0.8655	**0.9472**	**0.9888**	**0.9944**	0.6024	**0.1612**	0.5399
	θ_Q^L	**0.7689**	0.**7850**	**0.6533**	**0.8682**	0.9309	**0.9888**	**0.9944**	**0.7137**	**0.1612**	**0.5493**
Q+G	θ_Q	0.7604	0.7009	0.6893	0.8908	**0.9431**	**0.9888**	**0.9944**	0.5825	0.2500	0.6291
	θ_Q^G	**0.8160**	**0.7423**	**0.7867**	**0.9188**	0.9106	0.9372	0.9796	**0.6600**	**0.5658**	**0.7653**
Q+L+G	θ_Q	0.7800	0.7583	0.6893	0.8921	**0.9431**	**0.9888**	**0.9944**	0.6680	0.2500	0.6338
	θ_Q^{L+G}	**0.8516**	**0.8278**	**0.7867**	**0.9401**	0.8821	0.9372	0.9814	**0.8012**	**0.5658**	**0.8357**

Table 5: Comparison between the performance of θ_Q and expanded query language models in terms of micro average accuracy. θ_{N_E} was used in ranking.

On the other hand, again we observe that the effects on the Nil and the non-Nil queries are different. While in Table 4 the alternative name strings do not affect the performance much for Nil queries, now the expanded query language models actually hurt the performance for Nil queries. It is not surprising to see this result. When we expand the query language model, we can possibly introduce noise, especially when we use the external knowledge obtained from Wikipedia, which largely depends on what Wikipedia considers to be the most popular official name of a query name string. With noisy terms in the expanded query language model we increase the chance to link the query to a KB entry which is not the correct match. The challenge is that we do not know when additional terms in the expanded query language model are noise and when they are not, because for non-Nil queries we do observe a substantial amount of improvement brought by query expansion, especially with external world knowledge. We will further investigate this research question in the future.

We now further study the impact of using the KB disambiguation text associated with each entry to estimate the KB entry language model used in the KL-divergence ranking function. The results are shown in Table 6 for all the methods on θ_{N_E} vs. $\theta_{N_E+D_E}$ using the expanded query language models. We can see that for all methods the impact of using the KB disambiguation text is very minimal and is observed only for GPE and ORG queries. Table 7 shows an example of the KL-divergence scores for a query, *Mobile* whose context is previously shown in the Figure 1. Without the KB disambiguation text both the KB entry *Mobile Alabama* and the entry *Mobile River* are given the same score, resulting in inaccurate linking in the θ_{N_E} case. But with $\theta_{N_E+D_E}$, *Mobile Alabama* was scored higher, resulting in an accurate linking. However, we observe that such cases are very rare in the TAC 2010 query list and thus the overall improvement observed is minimal.

KB Entry	KB Name	w/o text	w/ text
E0583976	Mobile Alabama	-6.28514	-6.3839
E0183287	Mobile River	-6.28514	-6.69372

Table 7: The KL-divergence scores of KB entities for the query *Mobile*.

Finally, we compare our performance with the highest scores from TAC-KBP 2010 as shown in the Table 8. It is important to note that the highest TAC results shown in the table under each setting are not necessarily obtained by the same team. We can see that our overall performance when KB text is used is competitive compared with the highest TAC score, and is substantially higher than the TAC score when KB text is not used. Lehmann et al. (2010) achieved highest TAC scores. They used a variety of evidence from Wikipedia like disambiguation pages, anchors, expanded acronyms and redirects to build a rich feature set. But as we discussed, building a rich fea-

Method	KB Text	All				Nil			Non-Nil		
		ALL	GPE	ORG	PER	GPE	ORG	PER	GPE	ORG	PER
Q	θ_{N_E}	**0.6916**	**0.5714**	0.6533	0.8495	0.8618	0.9888	0.9963	0.4294	0.1612	0.4789
	$\theta_{N_E+D_E}$	0.6888	0.5607	0.6533	0.8495	0.8618	0.9888	0.9963	0.4135	0.1612	0.4789
Q+L	θ_{N_E}	0.7689	0.7850	0.6533	0.8682	0.9309	0.9888	0.9944	0.7137	0.1612	0.5493
	$\theta_{N_E+D_E}$	**0.7707**	**0.7904**	0.6533	0.8682	**0.9390**	0.9888	0.9944	**0.7177**	0.1612	0.5493
Q+G	θ_{N_E}	0.8160	0.7423	**0.7867**	0.9188	**0.9106**	0.9372	0.9796	0.6600	**0.5658**	0.7653
	$\theta_{N_E+D_E}$	**0.8222**	**0.7450**	0.7827	**0.9387**	0.8902	0.9372	**0.9814**	**0.6740**	0.5559	**0.8310**
Q+L+G	θ_{N_E}	0.8516	0.8278	0.7867	0.9401	**0.8821**	0.9372	0.9814	0.8012	0.5658	0.8357
	$\theta_{N_E+D_E}$	**0.8524**	**0.8291**	**0.7880**	0.9401	0.8740	0.9372	0.9814	**0.8072**	**0.5691**	0.8357

Table 6: Comparing the performance using KB text and without using KB text for all methods using expanded query models in terms of micro average accuracy on 2250 queries. $\theta_{N_E+D_E}$ represents method using KB text and θ_{N_E} represents methods without using KB text.

ture set is an expensive task. Their overall accuracy is 1.5% higher than our model. Table 8 shows that the performance of ORG entities is lower when compared with the TAC results when we used KB text. In our analysis, we observed that, even though some entities like AMPAS are linked correctly, the entities like CCC (Consolidated Contractors Company) failed due to ambiguity in the title. Here, we may benefit by leveraging more global knowledge, i.e, we should expand the N_Q^g with Wikipedia global context entities together with the title to fully benefit from global knowledge. In particular, when KB text is not used, our system outperforms the highest TAC results for all three types of queries.

From the analysis by Ji et al. (2010), overall the participating teams generally performed the best on PER queries and the worst on GPE queries. With our system, we can achieve good performance on GPE queries.

KB Text Usage	Type	Our System	TAC Highest
$\theta_{N_E+D_E}$	All	0.8524	**0.8680**
	GPE	**0.8291**	0.7957
	ORG	0.7880	**0.8520**
	PER	0.9401	**0.9601**
θ_{N_E}	All	**0.8516**	0.7791
	GPE	**0.8278**	0.7076
	ORG	**0.7867**	0.7333
	PER	**0.9401**	0.9001

Table 8: Comparison of the best configuration of our system (Q+L+G with θ_Q^{L+G}) with the TAC-KBP 2010 results in terms of micro-averaged accuracy. $\theta_{N_E+D_E}$ represents the method using KB disambiguation text and θ_{N_E} represents the method without using KB disambiguation text.

4.3 Parameter Sensitivity

In all our experiments, we set the Dirichlet prior μ to 2500 following previous studies. For the threshold τ we empirically set it to -12.0 in all the experiments based on preliminary results. Recall that all the expanded query language models also have a control parameters α. The local context-based models θ_Q^L and θ_Q^{L+G} have an additional parameter σ which controls the proximity weighing. The θ_Q^{L+G} model has another additional parameter β that controls the balance between local context and world knowledge. In this subsection, we study the sensitivity of these parameters. We plot the sensitivity graphs for all the methods that involve α (β set to 0.5) in Figure 3. As we can see, all the curves appear to be stable and α=0.4 appears to work well.

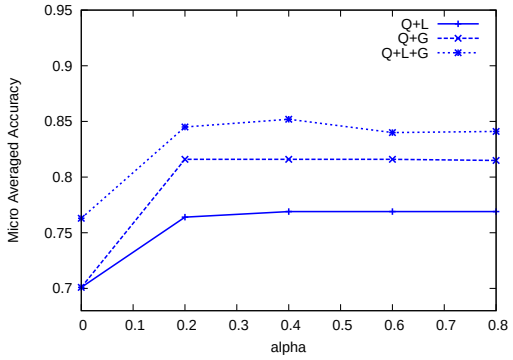

Figure 3: Sensitivity of α in regard to micro-averaged accuracy.

Similarly, we set α=0.4 and examine how β affects micro averaged accuracy. We plot the sensitivity curve for β for the Q+L+G setting with θ_Q^{L+G} in Figure 5. As we can see, the best performance is achieved when β=0.5. This implies that the local

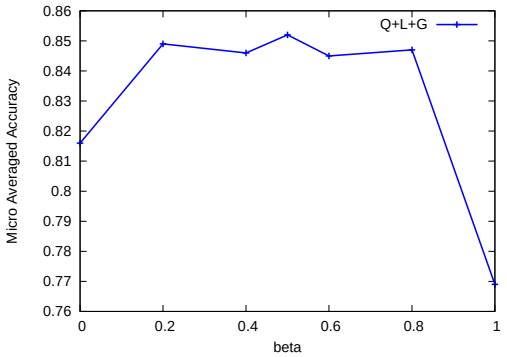

Figure 4: Sensitivity of β in regard to micro-averaged accuracy.

context and the global world knowledge are weighed equally for aiding disambiguation and improving the entity linking performance.

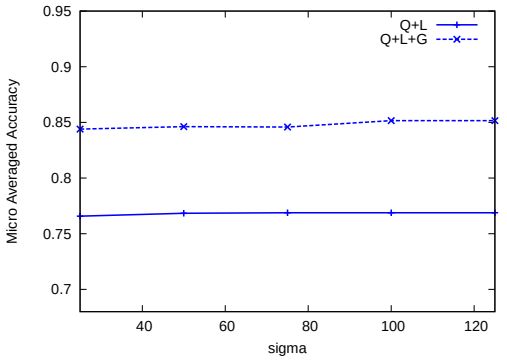

Figure 5: Sensitivity of σ with respect to micro-averaged accuracy.

Furthermore, we systematically test a fixed set of σ values from 25 to 125 with an intervals of 25 and examine how σ affects micro averaged accuracy. We set α=0.4 and β=0.5, which is the best parameter setting as discussed above. We plot the sensitivity curves for the parameter σ for methods that utilize the local context, i.e. θ_Q^L and θ_Q^{L+G}, in Figure 5. We observe that all the curves are stable and 75 $<=$ σ $<=$ 100 appears to work well. We set σ=100 for all our experiments. Moreover, after 100, the graph becomes stable, which indicates that proximity has less impact on the method from this point on. This implies that an equal weighing scheme actually would work the same for these experiments. Part of the reason may be that by using only named entities in the context rather than all words, we have effectively picked the most useful contextual terms. Therefore, positional feedback models do have exhibit much benefit for our problem.

5 Related Work

Bunescu and Pasca (2006) and Cucerzan (2007) explored the entity linking task using Vector Space Models for ranking. They took a classification approach together with the novel idea of exploiting Wikipedia knowledge. In their pioneering work, they used Wikipedia's category information for entity disambiguation. They show that using different background knowledge, we can find efficient approaches for disambiguation. In their work, they took an assumption that every entity has a KB entry and thus the NIL entries are not handled.

Similar to other researchers, Zhang et al. (2010) took an approach of classification and used a two-stage approach for entity liking. They proposed a supervised model with SVM ranking to filter out the candidates and deal with disambiguation effectively. For entity diambiguation they used the contextual comparisons between the Wikipedia article and the KB article. However, their work ignores the possibilities of acronyms in the entities. Also, the ambiguous geo-political names are not handled in their work.

Dredze et al. (2010) took the approach that large number of entities will be unlinkable, as there is a probability that the relevant KB entry is unavailable. Their algorithm for learning NIL has shown very good results. But their proposal for handling the alias name or stage name via multiple lists is not scalable. Unlike their approach, we use the global knowledge to handle the stage names and thus this gives an optimized solution to handle alias names. Similarly, for acronyms we use the global knowledge that aids unabbreviating and thus entity disambiguation. Similar to other approaches, Zheng et al. (2010) took a learning to rank approach and compared list-wise rank model to the pair-wise rank model. They achieved good results on the list-wise ranking approach. They handled acronyms and disambiguity through wiki redirect pages and the anchor texts which is similar to others ideas.

Challenges in supervised learning includes careful feature selection. The features can be selected in ad hoc manner - similarity based or semantic based. Also machine learning approach induces challenges of handling heterogenous cases. Unlike their machine learning approach which requires careful fea-

ture engineering and heterogenous training data, our method is simple as we use simple similarity measures. At the same time, we propose a statistical language modeling approach to the linking problem. Many researchers have proposed efficient ideas in their works. We integrated some of their ideas like world knowledge with our new techniques to achieve efficient entity linking accuracy.

6 Conclusions

In this paper we proposed a novel approach to entity linking based on statistical language model-based information retrieval with query expansion using the local context from the query document as well as world knowledge from the Web. Our model is a simple unsupervised one that follows principled existing information retrieval techniques. And yet it performs the entity linking task effectively compared with the best performance achieved in the TAC-KBP 2010 evaluation.

Currently our model does not exploit world knowledge from the Web completely. World knowledge, especially obtained from Wikipedia, has shown to be useful in previous studies. As our future work, we plan to explore how to further incorporate such world knowledge into our model in a principled way.

References

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceesings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 9–16, Trento, Italy.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In *Proceedings of the Third Text Analysis Conference*.

John Lafferty and ChengXiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 111–119.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127.

John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. 2010. Lcc approaches to knowledge base population at tac 2010. In *Proceedings TAC 2010 Workshop*. TAC 2010.

Yuanhua Lv and ChengXiang Zhai. 2009. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pages 1895–1898.

Yuanhua Lv and ChengXiang Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In *Proceeding of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 579–586.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the TAC 2009 knowledge base population track. In *Proceedings of the Second Text Analysis Conference*.

ChengXiang Zhai and John Lafferty. 2001. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 403–410.

Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, April.

Wei Zhang, Jian Su, Chew Lim Tan, and Wen Ting Wang. 2010. Entity linking leveraging automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1290–1298.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491.