Word Sense Disambiguation Improves Information Retrieval

Zhi Zhong and **Hwee Tou Ng**
Department of Computer Science
National University of Singapore
13 Computing Drive, Singapore 117417
{zhongzhi, nght}@comp.nus.edu.sg

Abstract

Previous research has conflicting conclusions on whether word sense disambiguation (WSD) systems can improve information retrieval (IR) performance. In this paper, we propose a method to estimate sense distributions for short queries. Together with the senses predicted for words in documents, we propose a novel approach to incorporate word senses into the language modeling approach to IR and also exploit the integration of synonym relations. Our experimental results on standard *TREC* collections show that using the word senses tagged by a supervised WSD system, we obtain significant improvements over a state-of-the-art IR system.

1 Introduction

Word sense disambiguation (WSD) is the task of identifying the correct meaning of a word in context. As a basic semantic understanding task at the lexical level, WSD is a fundamental problem in natural language processing. It can be potentially used as a component in many applications, such as machine translation (MT) and information retrieval (IR).

In recent years, driven by Senseval/Semeval workshops, WSD systems achieve promising performance. In the application of WSD to MT, research has shown that integrating WSD in appropriate ways significantly improves the performance of MT systems (Chan et al., 2007; Carpuat and Wu, 2007).

In the application to IR, WSD can bring two kinds of benefits. First, queries may contain ambiguous words (terms), which have multiple meanings. The ambiguities of these query words can hurt retrieval precision. Identifying the correct meaning of the ambiguous words in both queries and documents can help improve retrieval precision. Second, query words may have tightly related meanings with other words not in the query. Making use of these relations between words can improve retrieval recall.

Overall, IR systems can potentially benefit from the correct meanings of words provided by WSD systems. However, in previous investigations of the usage of WSD in IR, different researchers arrived at conflicting observations and conclusions. Some of the early research showed a drop in retrieval performance by using word senses (Krovetz and Croft, 1992; Voorhees, 1993). Some other experiments observed improvements by integrating word senses in IR systems (Schütze and Pedersen, 1995; Gonzalo et al., 1998; Stokoe et al., 2003; Kim et al., 2004).

This paper proposes the use of word senses to improve the performance of IR. We propose an approach to annotate the senses for short queries. We incorporate word senses into the language modeling (LM) approach to IR (Ponte and Croft, 1998), and utilize sense synonym relations to further improve the performance. Our evaluation on standard *TREC*[1] data sets shows that supervised WSD outperforms two other WSD baselines and significantly improves IR.

The rest of this paper is organized as follows. In Section 2, we first review previous work using WSD in IR. Section 3 introduces the LM approach to IR, including the pseudo relevance feedback method. We describe our WSD system and the method of

[1] http://trec.nist.gov/

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 273–282,
Jeju, Republic of Korea, 8-14 July 2012.

generating word senses for query terms in Section 4, followed by presenting our novel method of incorporating word senses and their synonyms into the LM approach in Section 5. We present experiments and analyze the results in Section 6. Finally, we conclude in Section 7.

2 Related Work

Many previous studies have analyzed the benefits and the problems of applying WSD to IR. Krovetz and Croft (1992) studied the sense matches between terms in query and the document collection. They concluded that the benefits of WSD in IR are not as expected because query words have skewed sense distribution and the collocation effect from other query terms already performs some disambiguation. Sanderson (1994; 2000) used pseudowords to introduce artificial word ambiguity in order to study the impact of sense ambiguity on IR. He concluded that because the effectiveness of WSD can be negated by inaccurate WSD performance, high accuracy of WSD is an essential requirement to achieve improvement. In another work, Gonzalo *et al.* (1998) used a manually sense annotated corpus, SemCor, to study the effects of incorrect disambiguation. They obtained significant improvements by representing documents and queries with accurate senses as well as synsets (synonym sets). Their experiment also showed that with the synset representation, which included synonym information, WSD with an error rate of 40%–50% can still improve IR performance. Their later work (Gonzalo et al., 1999) verified that part of speech (POS) information is discriminatory for IR purposes.

Several works attempted to disambiguate terms in both queries and documents with the senses predefined in hand-crafted sense inventories, and then used the senses to perform indexing and retrieval. Voorhees (1993) used the hyponymy ("IS-A") relation in WordNet (Miller, 1990) to disambiguate the polysemous nouns in a text. In her experiments, the performance of sense-based retrieval is worse than stem-based retrieval on all test collections. Her analysis showed that inaccurate WSD caused the poor results.

Stokoe *et al.* (2003) employed a fine-grained WSD system with an accuracy of 62.1% to disambiguate terms in both the text collections and the queries in their experiments. Their evaluation on TREC collections achieved significant improvements over a standard term based vector space model. However, it is hard to judge the effect of word senses because of the overall poor performances of their baseline method and their system.

Instead of using fine-grained sense inventory, Kim *et al.* (2004) tagged words with 25 root senses of nouns in WordNet. Their retrieval method maintained the stem-based index and adjusted the term weight in a document according to its sense matching result with the query. They attributed the improvement achieved on TREC collections to their coarse-grained, consistent, and flexible sense tagging method. The integration of senses into the traditional stem-based index overcomes some of the negative impact of disambiguation errors.

Different from using predefined sense inventories, Schütze and Pedersen (1995) induced the sense inventory directly from the text retrieval collection. For each word, its occurrences were clustered into senses based on the similarities of their contexts. Their experiments showed that using senses improved retrieval performance, and the combination of word-based ranking and sense-based ranking can further improve performance. However, the clustering process of each word is a time consuming task. Because the sense inventory is collection dependent, it is also hard to expand the text collection without re-doing preprocessing.

Many studies investigated the expansion effects by using knowledge sources from thesauri. Some researchers achieved improvements by expanding the disambiguated query words with synonyms and some other information from WordNet (Voorhees, 1994; Liu et al., 2004; Liu et al., 2005; Fang, 2008). The usage of knowledge sources from WordNet in document expansion also showed improvements in IR systems (Cao et al., 2005; Agirre et al., 2010).

The previous work shows that the WSD errors can easily neutralize its positive effect. It is important to reduce the negative impact of erroneous disambiguation, and the integration of senses into traditional term index, such as stem-based index, is a possible solution. The utilization of semantic relations has proved to be helpful for IR. It is also interest-

ing to investigate the utilization of semantic relations among senses in IR.

3 The Language Modeling Approach to IR

This section describes the LM approach to IR and the pseudo relevance feedback approach.

3.1 The language modeling approach

In the language modeling approach to IR, language models are constructed for each query q and each document d in a text collection C. The documents in C are ranked by the distance to a given query q according to the language models. The most commonly used language model in IR is the unigram model, in which terms are assumed to be independent of each other. In the rest of this paper, language model will refer to the unigram language model.

One of the commonly used measures of the similarity between query model and document model is negative Kullback-Leibler (KL) divergence (Lafferty and Zhai, 2001). With unigram model, the negative KL-divergence between model θ_q of query q and model θ_d of document d is calculated as follows:

$$\begin{aligned}-D(\theta_q||\theta_d) &= -\sum_{t\in V} p(t|\theta_q)\log\frac{p(t|\theta_q)}{p(t|\theta_d)}\\ &= \sum_{t\in V} p(t|\theta_q)\log p(t|\theta_d) - \sum_{t\in V} p(t|\theta_q)\log p(t|\theta_q)\\ &= \sum_{t\in V} p(t|\theta_q)\log p(t|\theta_d) + E(\theta_q), \qquad (1)\end{aligned}$$

where $p(t|\theta_q)$ and $p(t|\theta_d)$ are the generative probabilities of a term t from the models θ_q and θ_d, V is the vocabulary of C, and $E(\theta_q)$ is the entropy of q.

Define $tf(t,d)$ and $tf(t,q)$ as the frequencies of t in d and q, respectively. Normally, $p(t|\theta_q)$ is calculated with maximum likelihood estimation (MLE):

$$p(t|\theta_q) = \frac{tf(t,q)}{\sum_{t'\in q} tf(t',q)}. \qquad (2)$$

In the calculation of $p(t|\theta_d)$, several smoothing methods have been proposed to overcome the data sparseness problem of a language model constructed from one document (Zhai and Lafferty, 2001b). For example, $p(t|\theta_d)$ with the Dirichlet-prior smoothing can be calculated as follows:

$$p(t|\theta_d) = \frac{tf(t,d) + \mu\, p(t|\theta_C)}{\sum_{t'\in V} tf(t',d) + \mu}, \qquad (3)$$

where μ is the prior parameter in the Dirichlet-prior smoothing method, and $p(t|\theta_C)$ is the probability of t in C, which is often calculated with MLE:

$$p(t|\theta_C) = \frac{\sum_{d'\in C} tf(t,d')}{\sum_{d'\in C}\sum_{t'\in V} tf(t',d')}.$$

3.2 Pseudo relevance feedback

Pseudo relevance feedback (PRF) is widely used in IR to achieve better performance. It is constructed with two retrieval steps. In the first step, ranked documents are retrieved from C by a normal retrieval method with the original query q. In the second step, a number of terms are selected from the top k ranked documents D_q for query expansion, under the assumption that these k documents are relevant to the query. Then, the expanded query is used to retrieve the documents from C.

There are several methods to select expansion terms in the second step (Zhai and Lafferty, 2001a). For example, in Indri[2], the terms are first ranked by the following score:

$$v(t, D_q) = \sum_{d\in D_q} \log\left(\frac{tf(t,d)}{|d|} \times \frac{1}{p(t|\theta_C)}\right),$$

as in Ponte (1998). Define $p(q|\theta_d)$ as the probability score assigned to d. The top m terms T_q are selected with weights calculated based on the relevance model described in Lavrenko and Croft (2001):

$$w(t, D_q) = \sum_{d\in D_q} \left[\frac{tf(t,d)}{|d|} \times p(q|\theta_d) \times p(\theta_d)\right],$$

which calculates the sum of weighted probabilities of t in each document. After normalization, the probability of t in θ_q^r is calculated as follows:

$$p(t|\theta_q^r) = \frac{w(t,D_q)}{\sum_{t'\in T_q} w(t',D_q)}.$$

Finally, the relevance model is interpolated with the original query model:

$$p(t|\theta_q^{prf}) = \lambda\, p(t|\theta_q^r) + (1-\lambda) p(t|\theta_q), \qquad (4)$$

where parameter λ controls the amount of feedback. The new model θ_q^{prf} is used to replace the original one θ_q in Equation 1.

Collection enrichment (CE) (Kwok and Chan, 1998) is a technique to improve the quality of the feedback documents by making use of an external target text collection X in addition to the original target C in the first step of PRF. The usage of X is supposed to provide more relevant feedback documents and feedback query terms.

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

4 Word Sense Disambiguation

In this section, we first describe the construction of our WSD system. Then, we propose the method of assigning senses to query terms.

4.1 Word sense disambiguation system

Previous research shows that translations in another language can be used to disambiguate the meanings of words (Chan and Ng, 2005; Zhong and Ng, 2009). We construct our supervised WSD system directly from parallel corpora.

To generate the WSD training data, 7 parallel corpora were used, including *Chinese Treebank*, *FBIS Corpus*, *Hong Kong Hansards*, *Hong Kong Laws*, *Hong Kong News*, *Sinorama News Magazine*, and *Xinhua Newswire*. These corpora were already aligned at sentence level. We tokenized English texts with *Penn Treebank Tokenizer*, and performed word segmentation on Chinese texts. Then, word alignment was performed on the parallel corpora with the *GIZA++* software (Och and Ney, 2003).

For each English morphological root e, the English sentences containing its occurrences were extracted from the word aligned output of GIZA++, as well as the corresponding translations of these occurrences. To minimize noisy word alignment result, translations with no Chinese character were deleted, and we further removed a translation when it only appears once, or its frequency is less than 10 and also less than 1% of the frequency of e. Finally, only the most frequent 10 translations were kept for efficiency consideration.

The English part of the remaining occurrences were used as training data. Because multiple English words may have the same Chinese translation, to differentiate them, each Chinese translation is concatenated with the English morphological root to form a word sense. We employed a supervised WSD system, *IMS*[3], to train the WSD models. IMS (Zhong and Ng, 2010) integrates multiple knowledge sources as features. We used MaxEnt as the machine learning algorithm. Finally, the system can disambiguate the words by assigning probabilities to different senses.

[3] http://nlp.comp.nus.edu.sg/software/ims

4.2 Estimating sense distributions for query terms

In IR, both terms in queries and the text collection can be ambiguous. Hence, WSD is needed to disambiguate these ambiguous terms. In most cases, documents in a text collection are full articles. Therefore, a WSD system has sufficient context to disambiguate the words in the document. In contrast, queries are usually short, often with only two or three terms in a query. Short queries pose a challenge to WSD systems since there is insufficient context to disambiguate a term in a short query.

One possible solution to this problem is to find some text fragments that contain a query term. Suppose we already have a basic IR method which does not require any sense information, such as the stem-based LM approach. Similar to the PRF method, assuming that the top k documents retrieved by the basic method are relevant to the query, these k documents can be used to represent query q (Broder et al., 2007; Bendersky et al., 2010; He and Wu, 2011). We propose a method to estimate the sense probabilities of each query term of q from these top k retrieved documents.

Suppose the words in all documents of the text collection are disambiguated with a WSD system, and each word occurrence w in document d is assigned a vector of senses, $S(w)$. Define the probability of assigning sense s to w as $p(w, s, d)$. Given a query q, suppose D_q is the set of top k documents retrieved by the basic method, with the probability score $p(q|\theta_d)$ assigned to $d \in D_q$.

Given a query term $t \in q$
$S(t, q) = \{\}$
$sum = 0$
for each document $d \in D_q$
  for each word occurrence $w \in d$, whose stem form is identical to the stem form of t
    for each sense $s \in S(w)$
      $S(t, q) = S(t, q) \cup \{s\}$
      $p(t, s, q) = p(t, s, q) + p(q|\theta_d)\ p(w, s, d)$
      $sum = sum + p(q|\theta_d)\ p(w, s, d)$
for each sense $s \in S(t, q)$
  $p(t, s, q) = p(t, s, q)/sum$
Return $S(t, q)$, with probability $p(t, s, q)$ for $s \in S(t, q)$

Figure 1: Process of generating senses for query terms

Figure 1 shows the pseudocode of calculating the

sense distribution for a query term t in q with D_q, where $S(t,q)$ is the set of senses assigned to t and $p(t,s,q)$ is the probability of tagging t as sense s. Basically, we utilized the sense distribution of the words with the same stem form in D_q as a proxy to estimate the sense probabilities of a query term. The retrieval scores are used to weight the information from the corresponding retrieved documents in D_q.

5 Incorporating Senses into Language Modeling Approaches

In this section, we propose to incorporate senses into the LM approach to IR. Then, we describe the integration of sense synonym relations into our model.

5.1 Incorporating senses as smoothing

With the method described in Section 4.2, both the terms in queries and documents have been sense tagged. The next problem is to incorporate the sense information into the language modeling approach.

Suppose $p(t,s,q)$ is the probability of tagging a query term $t \in q$ as sense s, and $p(w,s,d)$ is the probability of tagging a word occurrence $w \in d$ as sense s. Given a query q and a document d in text collection C, we want to re-estimate the language models by making use of the sense information assigned to them.

Define the frequency of s in d as:

$$stf(s,d) = \sum_{w \in d} p(w,s,d),$$

and the frequency of s in C as:

$$stf(s,C) = \sum_{d \in C} stf(s,d).$$

Define the frequencies of sense set S in d and C as:

$$stf(S,d) = \sum_{s \in S} stf(s,d),$$
$$stf(S,C) = \sum_{s \in S} stf(s,C).$$

For a term $t \in q$, with senses $S(t,q)$:$\{s_1,...,s_n\}$, suppose V:$\{p(t,s_1,q),...,p(t,s_n,q)\}$ is the vector of probabilities assigned to the senses of t and W:$\{stf(s_1,d),...,stf(s_n,d)\}$ is the vector of frequencies of $S(t,q)$ in d. The function $\cos(t,q,d)$ calculates the cosine similarity between vector V and vector W. Assume D is a set of documents in C which contain any sense in $S(t,q)$, we define function $\overline{cos}(t,q) = \sum_{d \in D} cos(t,q,d)/|D|$, which calculates the mean of the sense cosine similarities, and define function $\Delta cos(t,q,d) = cos(t,q,d) - \overline{cos}(t,q)$, which calculates the difference between $cos(t,q,d)$ and the corresponding mean value.

Given a query q, we re-estimate the term frequency of query term t in d with sense information integrated as smoothing:

$$tf_{sen}(t,d) = tf(t,d) + sen(t,q,d), \quad (5)$$

where function $sen(t,q,d)$ is a measure of t's sense information in d, which is defined as follows:

$$sen(t,q,d) = \alpha^{\Delta cos(t,q,d)} stf(S(t,q),d). \quad (6)$$

In $sen(t,q,d)$, the last item $stf(S(t,q),d)$ calculates the sum of the sense frequencies of t senses in d, which represents the amount of t's sense information in d. The first item $\alpha^{\Delta cos(t,q,d)}$ is a weight of the sense information concerning the relative sense similarity $\Delta cos(t,q,d)$, where α is a positive parameter to control the impact of sense similarity. When $\Delta cos(t,q,d)$ is larger than zero, such that the sense similarity of d and q according to t is above the average, the weight for the sense information is larger than 1; otherwise, it is less than 1. The more similar they are, the larger the weight value. For $t \notin q$, because the sense set $S(t,q)$ is empty, $stf(S(t,q),d)$ equals to zero and $tf_{sen}(t,d)$ is identical to $tf(t,d)$.

With sense incorporated, the term frequency is influenced by the sense information. Consequently, the estimation of probability of t in d becomes query specific:

$$p(t|\theta_d^{sen}) = \frac{tf_{sen}(t,d) + \mu\, p(t|\theta_C^{sen})}{\sum_{t' \in V} tf_{sen}(t',d) + \mu}, \quad (7)$$

where the probability of t in C is re-calculated as:

$$p(t|\theta_C^{sen}) = \frac{\sum_{d' \in C} tf_{sen}(t,d')}{\sum_{d' \in C} \sum_{t' \in V} tf_{sen}(t',d')}.$$

5.2 Expanding with synonym relations

Words usually have some semantic relations with others. Synonym relation is one of the semantic relations commonly used to improve IR performance. In this part, we further integrate the synonym relations of senses into the LM approach.

Suppose $R(s)$ is the set of senses having synonym relation with sense s. Define $S(q)$ as the set of senses of query q, $S(q) = \bigcup_{t \in q} S(t,q)$, and define $R(s,q)$=$R(s)$−$S(q)$. We update the frequency of a query term t in d by integrating the synonym relations as follows:

$$tf_{syn}(t,d) = tf_{sen}(t,d) + syn(t,q,d), \quad (8)$$

where $syn(t, q, d)$ is a function measuring the synonym information in d:

$$syn(t, q, d) = \sum_{s \in S(t)} \beta(s, q) p(t, s, q) stf(R(s, q), d).$$

The last item $stf(R(s, q), d)$ in $syn(t, q, d)$ is the sum of the sense frequencies of $R(s, q)$ in d. Notice that the synonym senses already appearing in $S(q)$ are not included in the calculation, because the information of these senses has been used in some other places in the retrieval function. The frequency of synonyms, $stf(R(s, q), d)$, is weighted by $p(t, s, q)$ together with a scaling function $\beta(s, q)$:

$$\beta(s, q) = \min(1, \frac{stf(s, C)}{stf(R(s, q), C)}).$$

When $stf(s, C)$, the frequency of sense s in C, is less than $stf(R(s, q), C)$, the frequency of $R(s, q)$ in C, the function $\beta(s, q)$ scales down the impact of synonyms according to the ratio of these two frequencies. The scaling function makes sure that the overall impact of the synonym senses is not greater than the original word senses.

Accordingly, we have the probability of t in d updated to:

$$p(t|\theta_d^{syn}) = \frac{tf_{syn}(t, d) + \mu\, p(t|\theta_C^{syn})}{\sum_{t' \in V} tf_{syn}(t', d) + \mu}, \quad (9)$$

and the probability of t in C is calculated as:

$$p(t|\theta_C^{syn}) = \frac{\sum_{d' \in C} tf_{syn}(t, d')}{\sum_{d' \in C} \sum_{t' \in V} tf_{syn}(t', d')}.$$

With this language model, the probability of a query term in a document is enlarged by the synonyms of its senses; The more its synonym senses in a document, the higher the probability. Consequently, documents with more synonym senses of the query terms will get higher retrieval rankings.

6 Experiments

In this section, we evaluate and analyze the models proposed in Section 5 on standard TREC collections.

6.1 Experimental settings

We conduct experiments on the TREC collection. The text collection C includes the documents from TREC disk 4 and 5, minus the CR (Congressional Record) corpus, with 528,155 documents in total. In addition, the other documents in TREC disk 1 to 5 are used as the external text collection X.

We use 50 queries from TREC6 Ad Hoc task as the development set, and evaluate on 50 queries from TREC7 Ad Hoc task, 50 queries from TREC8 Ad Hoc task, 50 queries from ROBUST 2003 (RB03), and 49 queries from ROBUST 2004 (RB04). In total, our test set includes 199 queries. We use the terms in the title field of TREC topics as queries. Table 1 shows the statistics of the five query sets. The first column lists the query topics, and the column *#qry* is the number of queries. The column *Ave* gives the average query length, and the column *Rels* is the total number of relevant documents.

Query Set	Topics	#qry	Ave	Rels
TREC6	301–350	50	2.58	4,290
TREC7	351–400	50	2.50	4,674
TREC8	401–450	50	2.46	4,728
RB03	601–650	50	3.00	1,658
RB04[4]	651–700	49	2.96	2,062

Table 1: Statistics of query sets

We use the *Lemur* toolkit (Ogilvie and Callan, 2001) version 4.11 as the basic retrieval tool, and select the default unigram LM approach based on KL-divergence and Dirichlet-prior smoothing method in Lemur as our basic retrieval approach. Stop words are removed from queries and documents using the standard INQUERY stop words list (Allan et al., 2000), and then the Porter stemmer is applied to perform stemming. The stem forms are finally used for indexing and retrieval.

We set the smoothing parameter μ in Equation 3 to 400 by tuning on TREC6 query set in a range of $\{100, 400, 700, 1000, 1500, 2000, 3000, 4000, 5000\}$. With this basic method, up to 10 top ranked documents D_q are retrieved for each query q from the extended text collection $C \cup X$, for the usage of performing PRF and generating query senses.

For PRF, we follow the implementation of Indri's PRF method and further apply the CE technique as described in Section 3.2. The number of terms selected from D_q for expansion is tuned from range $\{20, 25, 30, 35, 40\}$ and set to 25. The interpolation parameter λ in Equation 4 is set to 0.7 from range

[4]Topic 672 is eliminated, since it has no relevant document.

Method	TREC7	TERC8	RB03	RB04	Comb	Impr	#ret-rel
Top 1	0.2530	0.3063	0.3704	**0.4019**	-	-	-
Top 2	0.2488	0.2876	0.3065	0.4008	-	-	-
Top 3	0.2427	0.2853	0.3037	0.3514	-	-	-
Stem_{prf} (Baseline)	0.2634	0.2944	0.3586	0.3781	0.3234	-	9248
Stem_{prf}+MFS	0.2655	0.2971	$0.3626^{\dagger}$	0.3802	$0.3261^{\dagger}$	0.84%	9281
Stem_{prf}+Even	0.2655	0.2972	$0.3623^{\dagger}$	0.3814	$0.3263^{\ddagger}$	0.91%	9284
Stem_{prf}+WSD	$0.2679^{\ddagger}$	$0.2986^{\dagger}$	$0.3649^{\ddagger}$	0.3842	$0.3286^{\ddagger}$	1.63%	9332
Stem_{prf}+MFS+Syn	$0.2756^{\ddagger}$	$0.3034^{\dagger}$	$0.3649^{\dagger}$	0.3859	$0.3322^{\ddagger}$	2.73%	9418
Stem_{prf}+Even+Syn	$0.2713^{\dagger}$	$0.3061^{\ddagger}$	$0.3657^{\ddagger}$	$0.3859^{\dagger}$	$0.3320^{\ddagger}$	2.67%	9445
Stem_{prf}+WSD+Syn	**0.2762**‡	**0.3126**‡	**0.3735**‡	$0.3891^{\dagger}$	$0.3376^{\ddagger}$	4.39%	9538

Table 2: Results on test set in MAP score. The first three rows show the results of the top participating systems, the next row shows the performance of the baseline method, and the rest rows are the results of our method with different settings. Single dagger (†) and double dagger (‡) indicate statistically significant improvement over $\textit{Stem}_{prf}$ at the 95% and 99% confidence level with a two-tailed paired t-test, respectively. The best results are highlighted in bold.

$\{0.1, 0.2, ..., 0.9\}$. The CE-PRF method with this parameter setting is chosen as the baseline.

To estimate the sense distributions for terms in query q, the method described in Section 4.2 is applied with D_q. To disambiguate the documents in the text collection, besides the usage of the supervised WSD system described in Section 4.1, two WSD baseline methods, *Even* and *MFS*, are applied for comparison. The method *Even* assigns equal probabilities to all senses for each word, and the method *MFS* tags the words with their corresponding most frequent senses. The parameter α in Equation 6 is tuned on *TREC6* from 1 to 10 in increment of 1 for each sense tagging method. It is set to 7, 6, and 9 for the supervised WSD method, the Even method, and the MFS method, respectively.

Notice that the sense in our WSD system is conducted with two parts, a morphological root and a Chinese translation. The Chinese parts not only disambiguate senses, but also provide clues of connections among different words. Assume that the senses with the same Chinese part are synonyms, therefore, we can generate a set of synonyms for each sense, and then utilize these synonym relations in the method proposed in Section 5.2.

6.2 Experimental results

For evaluation, we use average precision (AP) as the metric to evaluate the performance on each query q:

$$\text{AP}(q) = \frac{\sum_{r=1}^{R}[p(r)rel(r)]}{relevance(q)},$$

where $relevance(q)$ is the number of documents relevant to q, R is the number of retrieved documents, r is the rank, $p(r)$ is the precision of the top r retrieved documents, and $rel(r)$ equals to 1 if the rth document is relevant, and 0 otherwise. Mean average precision (MAP) is a metric to evaluate the performance on a set of queries Q:

$$\text{MAP}(Q) = \frac{\sum_{q \in Q} \text{AP}(q)}{|Q|},$$

where $|Q|$ is the number of queries in Q.

We retrieve the top-ranked 1,000 documents for each query, and use the MAP score as the main comparing metric. In Table 2, the first four columns are the MAP scores of various methods on the TREC7, TREC8, RB03, and RB04 query sets, respectively. The column *Comb* shows the results on the union of the four test query sets. The first three rows list the results of the top three systems that participated in the corresponding tasks. The row $\textit{Stem}_{prf}$ shows the performance of our baseline method, the stem-based CE-PRF method. The column *Impr* calculates the percentage improvement of each method over the baseline $\textit{Stem}_{prf}$ in column *Comb*. The last column *#ret-rel* lists the total numbers of relevant documents retrieved by different methods.

The rows $\textit{Stem}_{prf}$+{*MFS*, *Even*, *WSD*} are the results of $\textit{Stem}_{prf}$ incorporating with the senses generated for the original query terms, by applying the approach proposed in Section 5.1, with the *MFS* method, the *Even* method, and our supervised WSD method, respectively. Comparing to the baseline method, all methods with sense integrated achieve consistent improvements on all query sets. The usage of the supervised WSD method outperforms the other two WSD baselines, and it achieves sta-

tistically significant improvements over $Stem_{prf}$ on TREC7, TREC8, and RB03.

The integration of senses into the baseline method has two aspects of impact. First, the morphological roots of senses conquer the irregular inflection problem. Thus, the documents containing the irregular inflections are retrieved when senses are integrated. For example, in topic 326 {*ferry sinkings*}, the stem form of *sinkings* is *sink*. As *sink* is an irregular verb, the usage of senses improves the retrieval recall by retrieving the documents containing the inflection forms *sunk*, *sank*, and *sunken*.

Second, the senses output by supervised WSD system help identify the meanings of query terms. Take topic 357 {*territorial waters dispute*} for example, the stem form of *waters* is *water* and its appropriate sense in this query should be water_水域 (body of water) instead of the most frequent sense of water_水 (H_2O). In $Stem_{prf}$+*WSD*, we correctly identify the minority sense for this query term. In another example, topic 425 {*counterfeiting money*}, the stem form of *counterfeiting* is *counterfeit*. Although the most frequent sense counterfeit_冒牌 (not genuine) is not wrong, another sense counterfeit_伪钞 (forged money) is more accurate for this query term. The Chinese translation in the latter sense represents the meaning of the phrase in original query. Thus, $Stem_{prf}$+*WSD* outperforms the other two methods on this query by assigning the highest probability for this sense.

Overall, the performance of $Stem_{prf}$+*WSD* is better than $Stem_{prf}$+{*MFS, Even*} on 121 queries and 119 queries, respectively. The *t-test* at the confidence level of 99% indicates that the improvements are statistically significant.

The results of expanding with synonym relations in the above three methods are shown in the last three rows, $Stem_{prf}$+{*MFS*, *Even*, *WSD*}+Syn. The integration of synonym relations further improves the performance no matter what kind of sense tagging method is applied. The improvement varies with different methods on different query sets. As shown in the last column of Table 2, the number of relevant documents retrieved is increased for each method. $Stem_{prf}$+*Even*+*Syn* retrieves more relevant documents than $Stem_{prf}$+*MFS*+*Syn*, because the former method expands more senses. Overall, the improvement achieved by $Stem_{prf}$+*WSD*+*Syn* is larger than the other two methods. It shows that the WSD technique can help choose the appropriate senses for synonym expansion.

Among the different settings, $Stem_{prf}$+*WSD*+*Syn* achieves the best performance. Its improvement over the baseline method is statistically significant at the 95% confidence level on RB04 and at the 99% confidence level on the other three query sets, with an overall improvement of 4.39%. It beats the best participated systems on three out of four query sets[5], including *TREC7, TREC8, and RB03*.

7 Conclusion

This paper reports successful application of WSD to IR. We proposed a method for annotating senses to terms in short queries, and also described an approach to integrate senses into an LM approach for IR. In the experiment on four query sets of TREC collection, we compared the performance of a supervised WSD method and two WSD baseline methods. Our experimental results showed that the incorporation of senses improved a state-of-the-art baseline, a stem-based LM approach with PRF method. The performance of applying the supervised WSD method is better than the other two WSD baseline methods. We also proposed a method to further integrate the synonym relations to the LM approaches. With the integration of synonym relations, our best performance setting with the supervised WSD achieved an improvement of 4.39% over the baseline method, and it outperformed the best participating systems on three out of four query sets.

Acknowledgments

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

References

E. Agirre, X. Arregi, and A. Otegi. 2010. Document expansion based on WordNet for robust IR. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 9–17.

[5]The top two systems on *RB04* are the results of the same participant with different configurations. They used lots of web resources, such as search engines, to improve the performance.

J. Allan, M. E. Connell, W.B. Croft, F.F. Feng, D. Fisher, and X. Li. 2000. INQUERY and TREC-9. In *Proceedings of the 9th Text REtrieval Conference*, pages 551–562.

M. Bendersky, W. B. Croft, and D. A. Smith. 2010. Structural annotation of search queries using pseudo-relevance feedback. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pages 1537–1540.

A. Broder, M. Fontoura, E. Gabrilovich, A. Joshi, V. Josifovski, and T. Zhang. 2007. Robust classification of rare queries using web knowledge. In *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 231–238.

G. Cao, J. Y. Nie, and J. Bai. 2005. Integrating word relationships into language models. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–305.

M. Carpuat and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 61–72.

Y. S. Chan and H. T. Ng. 2005. Scaling up word sense disambiguation via parallel texts. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 1037–1042.

Y. S. Chan, H. T. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 33–40.

H. Fang. 2008. A re-examination of query expansion using lexical resources. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies*, pages 139–147.

J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarrin. 1998. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING-ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pages 38–44.

J. Gonzalo, A. Penas, and F. Verdejo. 1999. Lexical ambiguity and information retrieval revisited. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 195–202.

D. He and D. Wu. 2011. Enhancing query translation with relevance feedback in translingual information retrieval. *Information Processing & Management*, 47(1):1–17.

S. B. Kim, H. C. Seo, and H. C. Rim. 2004. Information retrieval using word senses: root sense tagging approach. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 258–265.

R. Krovetz and W. B. Croft. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115–141.

K. L. Kwok and M. Chan. 1998. Improving two-stage ad-hoc retrieval for short queries. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 250–256.

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

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

S. Liu, F. Liu, C. Yu, and W. Meng. 2004. An effective approach to document retrieval via utilizing WordNet and recognizing phrases. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 266–272.

S. Liu, C. Yu, and W. Meng. 2005. Word sense disambiguation in queries. In *Proceedings of the 14th ACM Conference on Information and Knowledge Management*, pages 525–532.

G. A. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

P. Ogilvie and J. Callan. 2001. Experiments using the Lemur toolkit. In *Proceedings of the 10th Text REtrieval Conference*, pages 103–108.

J. M. Ponte and W. B. Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281.

J. M. Ponte. 1998. *A Language Modeling Approach to Information Retreival*. Ph.D. thesis, Department of Computer Science, University of Massachusetts.

M. Sanderson. 1994. Word sense disambiguation and information retrieval. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 142–151.

M. Sanderson. 2000. Retrieving with good sense. *Information Retrieval*, 2(1):49–69.

H. Schütze and J. O. Pedersen. 1995. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

C. Stokoe, M. P. Oakes, and J. Tait. 2003. Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 159–166.

E. M. Voorhees. 1993. Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of the 16th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–180.

E. M. Voorhees. 1994. Query expansion using lexical-semantic relations. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61–69.

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

C. Zhai and J. Lafferty. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342.

Z. Zhong and H. T. Ng. 2009. Word sense disambiguation for all words without hard labor. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1616–1621.

Z. Zhong and H. T. Ng. 2010. It Makes Sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics: System Demonstrations*, pages 78–83.