LSM: Language Sense Model for Information Retrieval

Shenghua Bao, Lei Zhang, Erdong Chen, Min Long, Rui Li, and Yong Yu

APEX Data and Knowledge Management Lab,
Department of Computer Science & Engineering,
Shanghai Jiao Tong University, Shanghai, P.R.China, 200230
{shhbao,zhanglei,edchen,mlong,rli,yyu}@apex.sjtu.edu.cn

Abstract. A lot of work has been done on drawing word senses into retrieval to deal with the word sense ambiguity problem, but most of them achieved negative results. In this paper, we first implement a WSD system for nouns and verbs, then the language sense model (LSM) for information retrieval is proposed. The LSM combines the terms and senses of a document seamlessly through an EM algorithm. Retrieval on TREC collections shows that the LSM outperforms both the vector space model and the traditional language model significantly for both medium and long queries (7.53%-16.90%). Based on the experiments, we can also empirically draw the conclusion that the fine-grained senses will improve the retrieval performance when they are properly used.

1 Introduction

Word sense disambiguation (WSD) has been studied for a long time in natural language processing. In the field of information retrieval (IR), word sense ambiguity is regarded as one of the main causes which affect the retrieval performance for two reasons:

- **Polysemy**: One word may have different meanings under different contexts.
- **Synonymy**: Different words may have the same meaning.

This encouraged various of work to integrate the WSD into IR. However, most of them achieved negative results, e.g. [1–3]. At the same time, the potential causes for the poor results were intensively studied, which can be categorized as follows:

- **Fine-grained Sense**: The words might be resolved to senses which are too specific for IR[4].
- **Poor WSD Results**: The low accuracy of the WSD system affects the final performance of the sense based IR system a lot[5].
- **Cannot Fall Back**: The pure sense based IR system can not fall back to the term based IR system. The term based IR system does not suffer from the ambiguity problem severely due to word collocation, and word senses' skewed distribution[6]. So It is not easy for a pure sense based IR system to exceed the term based IR system.

An example which successful solved the three problems above would be Kim's work in 2004 which improved retrieval performance significantly[7]. Firstly, it achieved the high WSD accuracy by using the coarse-grained senses which consisted of 25 root senses in WORDNET[1]. Secondly, it combined the root senses and the document terms through a revised vector space model. It successfully integrated the coarse-grained senses and terms. However, there are still some questions to be further addressed,

- Will the integration of fine-grained senses and terms work as well?
- Is there any other model to integrate the terms and senses besides the vector space model?

To answer these questions, we first implement a WSD system using the fine-grained senses in WORDNET. In our WSD system, only nouns and verbs are disambiguated because the nouns and verbs play important roles in IR and they are much easier to be disambiguated with a comparatively higher accuracy.

Then, the fine-grained senses are utilized based on the language model[8]. Firstly, the language model on term representation (LMTR) and sense representation (LMSR) are studied. Then, a novel model, language sense model (LSM) for information retrieval, is proposed which utilizes both sense and term representations. The experimental results on the TREC collections shows that the LMSR can not bring any improvement. However, the LSM outperforms both vector space model and traditional language model for medium and long queries significantly(7.53%-16.90%).

The rest of the paper is organized as follows. In Section 2, some related work is discussed . In Section 3, the process and evaluation of WSD are presented. The discussion of the LSM is given in Section 4. The experiment results of the LSM are given in Section 5. Finally, we make a conclusion and give some future work in Section 6.

2 Related Work

In this section, the related work is surveyed in two aspects. One relates to the previous efforts on WSD for IR while the other to the previous work on language models.

2.1 Word Sense Disambiguation for Information Retrieval

Most of the early work which integrated WSD into IR resulted in no improvement. A complete review of the integration of WSD and IR prior to the year 2000 can be found in the work of Sanderson [9]. In this section, we review some recent work which reported significant improvements by integrating WSD into IR.

In 2003, Stokoe represented documents and queries with sense vectors and retrieved the relevant documents using the traditional vector space model[10].Their experiments on TREC WT10G data collection empirically showed that their WSD system could significantly improve the retrieval performance. However, it was problematic that the absolute precision of the baseline and the proposed system were too low to investigate the effect of sense-based retrieval. Compared with Stokoe's work, the LSM improves the retrieval performance significantly when the baseline's absolute precision is comparatively much higher. More importantly, in the LSM, the terms and senses are integrated to achieve a better performance.

[1] http://wordnet.princeton.edu/

In 2004, Kim et al proposed the root sense tagging approach for information retrieval by integrating the root sense tags into the vector space model[7]. As proposed in Section 1, Kim solved the three existing problems successfully. Different from Kim's work, the LSM utilizes fine-grained senses, and combines the terms and senses in the language model.

2.2 Language Model

For many years, the primary consumers of statistical language models were speech recognition systems [11]. In 1998, Ponte and Croft [8] proposed a smoothed version of the document unigram model to assign a score to a query, which can be thought of as the probability that the query was generated from the document model. Since then, there emerged a great amount of research work related to language model. Most of them tried to solve the following two problems:

- **Data Sparseness**: Many smoothing methods were suggested to re-evaluate the probabilities of generating the query terms that did not appear in the document. Song and Croft proposed the good-tuning smoothing based on terms' power law distribution [12]. Zhai et al proposed the two-stage smoothing for language model[13]. In addition, cluster based smoothing methods were proposed and achieved significant improvement[14, 15]. In the LSM, the existing smoothing methods can be applied easily on both terms and senses to solve the data sparseness problem.
- **Term Dependency**: The unigram language model made an improper assumption that all terms were generated independently. Plenty of work has been done to model the proper dependencies between the query terms. Srikanth et al proposed the concept language model, where the query was viewed as a sequence of concepts and each concept as a sequence of terms[16]. Gao et al introduced the dependence language model by integrating the linkage of query terms as a hidden variable[17]. Recently, Cao et al exploited the word relations of WORDNET and co-occurrences and then integrated them into language models[18]. In the LSM, the query independent assumption can be relaxed to a certain extent as the terms and senses in the LSM depend strongly on each other.

3 Word Sense Disambiguation

Word Sense Disambiguation(WSD), which is a classical problem in Natural Language Processing(NLP), aims to improve the accuracy, namely the number of words correctly disambiguated. Our approach is based on the Co-occurrence, SEMCOR [2], and WORDNET. In order to achieve the high disambiguation accuracy, only the nouns and verbs on both the queries and the documents are disambiguated. Most of the methods are based on popular and effective techniques in [19, 7, 20].

SEMCOR2.0 is distributed with WORDNET2.0, an online thesaurus created at Princeton University. WORDNET2.0 consists of 90,000 terms and collocates organized into Synsets. Each Synset contains words which are synonymous with each other, while

[2] http://multisemcor.itc.it/semcor.php

the links between Synsets represent hypernymy and hyponomy relationships to form a WordNet hierarchical semantic network. Semcor2.0 is a manually sense tagged subset of the Brown Corpus consisting of 352 documents split into three data sets. The tag set used in Semcor consists of the unique sense identifiers used within WordNet.

At first, three pre-processing procedures are implemented. Firstly, each word is tagged with part-of-speech(POS) by Brill's tagger[3]. Secondly, ANNIE TAGGER[4] performs on the text to remove named entities from the WSD candidate set. In the experiment, only three types of named entities: LOC (location), PER (person) and ORG (organization) are extracted. Thirdly, each monosemous word is identified with the unique sense it owns. In the following three subsections, we will introduce the main methods of the WSD system.

Our WSD system makes use of mutual information(MI) of the adjacent words in the text. Besides, WordNet and Semcor information is integrated into the following procedures to identify the senses of the candidate words. We get context clues from the Semcor of the occurrence of the collocation. If, in all the occurrences of the collocation, the word has only one sense, and the number of the occurrences is larger than a given threshold($\geq$ 2 in our experiment), then we identify the word with the sense. We identify the sense of a word by comparing the original context of the word and the context set of the word's senses at WordNet and Semcor . The following nouns will be added to the context set of the sense: the words in the sense at WordNet, the first shortest noun phrase from the definition of the sense at WordNet, all the nouns which occur within a window size (20 words in our experiment)with respect to the sense in Semcor.

Our WSD system also integrates the hierarchical information of the synsets in WordNet. In WordNet, all the words with the same POS are organized into hierarchies, each synset is a part of a hierarchy. Taking the noun as an example, there are 25 root senses. For two words t_1 and t_2 within a window size, if the hierarchical distance between a sense of the word t_1 and the word t_2 is equal to or less than 1, the system identifies the two words with their corresponding senses.

We trained our method on the first 300 documents of Semcor, and tested it on the last 52 documents. The accuracy of noun is 78.12% and accuracy of verb is 60.58%, the overall accuracy of WSD system is 72.40%, which is much higher than the previous WSD sytem applied to IR.

4 Integrating Sense into Language Model

In this section, we talk about how to utilize the fine-grained word senses. In Section 4.1, the language model for term and sense representations is proposed. Then, the smoothing methods are discussed and a new hierarchical smoothing method is proposed. Finally, in Section 4.2, the LSM and the correspondingly parameter estimation methods are proposed to integrate the term and sense representations.

[3] http://www.cs.jhu.edu/ brill/
[4] http://gate.ac.uk/ie/annie.html

4.1 Language Model and Smoothing Methods

Language Model for Term and Sense Representations In this paper, we view that each document has two different presentations: namely term representation and sense representation, as shown in Figure 1. Two examples from TREC Fbis corpus are given in the right.

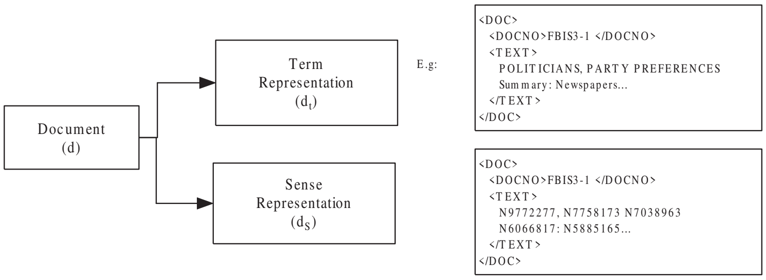

Fig. 1. Document Representations

The language model on term representations (LMTR) is the traditional approach. It first generates a model d_t for each document d. Given a query $q_t = q_{t1}q_{t2}\ldots q_{tm}$, the documents are ranked according to the probability the model could generate. In this paper, the urigram language model is adopted and the equation could be represented as follows:

$$P(q_t|d_t) = \prod_{i=1}^{m} P(q_{ti}|d_t) \quad (1)$$

Where q_t and d_t means the term representations of query q and document d respectively. q_{ti} means the ith term of the query q_t and m is the length of the query q_t.

The language model on sense representations (LMSR) is similar to the one on term representations. It first generates a sense model d_s for each document d using the sense representation, and then estimates the probability of d_s generating the sense query $q_s = q_{s1}q_{s2}\ldots q_{sm}$. The corresponding equation can be shown as follows.

$$P(q_s|d_s) = \prod_{i=1}^{m} P(q_{si}|d_s) \quad (2)$$

Smoothing Methods The smoothing method plays an important role in language model due to the data sparseness problem. An empirical study of smoothing methods for the language model can be found at [21]. Table 1 shows three of them which are popularly used in language model for information retrieval [13].

The three smoothing approaches can be applied to the LMTR and the LMSR. For the LMSR, we developed a new smoothing method, namely hierarchical smoothing,

Smoothing Methods	Formula
Jelinek-Mercer	$(1-\alpha)P(w\|d)+\alpha P(w\|C)$
Dirichlet	$\frac{c(w;d)+\mu P(w\|C)}{\sum_w c(w;d)+\mu}$
Absolute discount	$\frac{max(c(w;d)-\delta,0)}{\sum_w c(w;d)}+\frac{\delta\|d\|_u}{\|d\|}P(w\|C)$

Table 1. Smoothing Methods of Language Model for Information Retrieval

based on the WORDNET hierarchy as follows:

$$P^h(q_s|d_s)=\prod_{i=1}^{m}(1-\lambda_h)P(q_{si}|d_s)+\lambda_h P(Relative(q_{si})|d_s)) \tag{3}$$

Here the $Relative(q_{si})$ can be defined as the hypernym sense or hyponym sense of the sense q_{si} in the WORDNET hierarchy.λ_h is a constant from 0 to 1 which measures the confidence of the $Relative(q_{si})$.

4.2 Language Sense Model (LSM)

In this section, we firstly propose the language sense model (LSM) for information retrieval which utilizes both term and sense representations. Then the model parameter estimation is discussed.

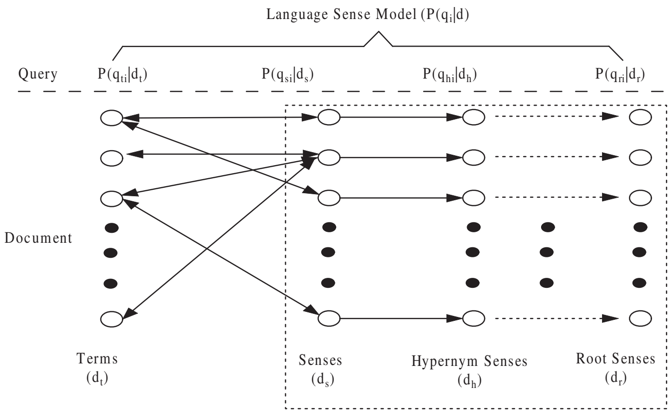

Fig. 2. Language Sense Model for Information Retrieval

Model Description Figure 2 shows the framework of the LSM. In the LSM, the model generates the probability of a given query from both document's term representation and sense representation. The sense representation d_s can be further extended to d_h and d_r which stand for sense's hypernym sense and root sense respectively. In this paper, we choose the d_s to be integrated with d_t as we want to study the effects of the fine-grained senses in information retrieval. So the LSM can be shown as Equation 4:

$$P(q|d) = \prod_{i=1}^{m}((1-\lambda)P(q_{ti}|d_t) + \lambda P(q_{si}|d_s)) \tag{4}$$

where $P(q_{ti}|d_t)$ and $P(q_{si}|d_s)$ means the probability of generating the ith query term q_i from term representation and sense representation respectively. Note that not all the terms in q_t can be disambiguated as the WSD is only conducted on the nouns and verbs. A default value will be given to the q_{si} if q_{ti} can not be disambiguated.

To solve the data sparseness problem, the existing smoothing method (as shown in Table 1) can be integrated into the LSM. An integration example of Jelinek-Mercer smoothing into LSM can be shown as follows:

$$P(q|d) = \prod_{i=1}^{m}(1-\alpha)[(1-\lambda)P(q_{ti}|d_t)+\lambda P(q_{si}|d_s)]+\alpha[(1-\lambda)P(q_{ti}|C_t)+\lambda P(q_{si}|C_s)] \tag{5}$$

Other than the traditional smoothing methods, the hierarchical smoothing can also be integrated by replacing $P(q_{si}|d_s)$ and $P(q_{si}|C_s)$ in Equation 5 with $P^h(q_{si}|d_s)$ and $P^h(q_{si}|C_s)$ defined in Equation 3 as follows:

$$P(q|d) = \prod_{i=1}^{m}(1-\alpha)[(1-\lambda)P(q_{ti}|d_t)+\lambda P^h(q_{si}|d_s)]+\alpha[(1-\lambda)P(q_{ti}|C_t)+\lambda P^h(q_{si}|C_s)] \tag{6}$$

Parameter Estimation To compute the query generating probability from the LSM, there are three components to be estimated: $P(q_{ti}|d_t)$, $P(q_{si}|d_s)$ and the combination parameter λ.

$P(q_{ti}|d_t)$, $P(q_{si}|d_s)$ can be estimated as the maximally likelihood of the term representation and sense representation generating the corresponding query term. Given a query, we estimate the optimal weights λ^* which could maximize the likelihood of the queries. This method is similar to Zhai's method in estimating the parameter of the two stage model[13] and Cao's method in estimating the combination in NSLM [18] . Let λ^* be the optimal weight, taking the formula 5 as an example, we have:

$$\lambda^* = \arg\max_{\lambda} \log \left\{ \begin{array}{c} (1-\alpha)\sum_{i=1}^{N}\pi_i \prod_{j=1}^{m}[(1-\lambda)P(q_{tj}|d_{ti}) + \lambda P(q_{sj}|d_{si})] \\ +\alpha \sum_{i=1}^{N}\pi_i \prod_{j=1}^{m}[(1-\lambda)P(q_{tj}|C_t) + \lambda P(q_{sj}|C_s)] \end{array} \right\} \tag{7}$$

where N is the number of documents in the dataset, and m is the length of query q. $\{\pi_i\}_{i=1}^{N}$ acts as the prior probability with which to choose the document to generate the query. With this setting, the EM formulae to update the parameter can be shown as follows:

$$\pi_i^{(r+1)} = \frac{\pi_i^{(r)} \prod_{j=1}^{m}[(1-\lambda^{(r)})P(q_{tj}|d_{ti}) + \lambda^{(r)}P(q_{sj}|d_{si})]}{\sum_{i=1}^{N} \pi_i^{(r)} \prod_{j=1}^{m}[(1-\lambda^{(r)})P(q_{tj}|d_{ti}) + \lambda^{(r)}P(q_{sj}|d_{si})]} \tag{8}$$

and

$$\lambda^{(r+1)} = \frac{1}{m}\sum_{j=1}^{m} \frac{(1-\alpha)\sum_{i=1}^{N} \pi_i^{(r)}\lambda^{(r)}P(q_{sj}|d_{si}) + \alpha\lambda^{(r)}P(q_{sj}|C_s)}{(1-\alpha)\sum_{i=1}^{N} \pi_i^{(r)}[(1-\lambda^{(r)})P(q_{tj}|d_{ti}) + \lambda^{(r)}P(q_{sj}|d_{si})] + \alpha[(1-\lambda^{(r)})P(q_{tj}|C_t) + \lambda^{(r)}P(q_{sj}|C_s)]} \tag{9}$$

The EM algorithm will be terminated if the log-likelihood of the query changes within a threshold. For Jelinek-Mercer smoothing, α is set to 0.4 in our experiments. The EM update formula for Dirichlet and Absolute Discount smoothing can be inferred similarly.

5 System Evaluation & Analysis

5.1 Experiment Setup

At first, all the nouns and verbs of queries and documents of TREC FBIS corpus were disambiguated with the methods proposed in Section 3. In order to evaluate the LSM's performance on different length queries, we generated three types of queries, shown as in Table 2.

Query Type	Query Count	Average Length(Term/Sense)	Extracted From
Short query	40	3.60 / 2.3	Title
Medium query	50	21.86 / 10.34	Tilte, Description
Long query	50	78.34 / 31.04	Title, Description,Narrative

Table 2. Short Queries, Medium Queries and Long Queries Extracted from the Routing Task

The LSM system is built based on the Lemur 3.1. In the following experiment, the standard mean average precision(MAP) and the total retrieved relevant document number(Recall) are used to evaluate the retrieval performance.

5.2 Evaluation of LMTR and LMSR

The results of language models on term and sense representations are compared on different queries and different smoothing methods, shown as Table 3. From the table, we can see that the language model on term representation (LMTR) performs much better than language model on sense representation (LMSR)in both precision and recall. Noting that some terms in the term representation cannot be disambiguated, we generated

Query Type	Smoothing Methods	LMTR (MAP/Recall)	LMSR (MAP/Recall)	Improved MAP	Improved Recall
Short Query	Jelinek-Mercer	0.1041 1692	0.0646 1388	-61.15%	-17.97%
	Dirichlet	0.1247 1859	0.0773 1569	-61.32%	-15.60%
	Absolute Discount	0.1133 1726	0.0736 1435	-53.94%	-16.86%
Medium Query	Jelinek-Mercer	0.1228 2329	0.0892 1887	-37.67%	-18.99%
	Dirichlet	0.1339 2357	0.1005 2126	-33.23%	-9.80%
	Absolute Discount	0.1150 2203	0.0961 1920	-19.67%	-12.85%
Long Query	Jelinek-Mercer	0.1649 2707	0.1222 2262	-34.94%	-16.07%
	Dirichlet	0.1630 2603	0.1337 2473	-21.91%	-4.99%
	Absolute Discount	0.1363 2431	0.1141 2142	-19.46%	-11.89%

Table 3. Comparison of LMTR and LMSR

Query Type	Smoothing Methods	LMTR (MAP/Recall)	LSM (MAP/Recall)	Improved MAP	Improved Recall	Sign MAP
Short Query	Jelinek-Mercer	0.1041 1692	0.1060 1677	1.90%	-0.89%	0.1695
	Dirichlet	0.1247 1859	0.1310 1897	5.06%	2.04%	0.1688
	Absolute Discount	0.1133 1726	0.1208 1769	6.59%	2.49%	0.1693
	Vector Space Model	0.1161 2042	◇ 0.1310 1897	12.83%	-7.10%	0.0506
Medium Query	Jelinek-Mercer	0.1228 2329	0.1344 2356	9.46%	1.16%	0.0805
	Dirichlet	0.1339 2357	0.1478 2492	10.36%	5.72%	0.0272
	Absolute Discount	0.1150 2203	0.1344 2326	16.90%	5.58%	* 0.0179
	Vector Space Model	0.1112 2391	◇ 0.1478 2492	32.91%	4.22%	* 0.0223
Long Query	Jelinek-Mercer	0.1649 2707	0.1792 2784	8.68%	2.84%	* 0.0381
	Dirichlet	0.1630 2603	0.1752 2719	7.53%	4.46%	* 0.0388
	Absolute Discount	0.1363 2431	0.1516 2526	11.26%	3.90%	* 0.0487
	Vector Space Model	0.0907 2531	◇ 0.1752 2719	93.16%	7.42%	* 0.0001

Table 4. Comparison of LMTR and LSM

a mixed document representation, where the undisambiguated terms are reserved in the sense representation. However, we got the conclusion again that the LMTR performs much better than the language model on mixed representations.

In addition, the hierarchical smoothing for LMSR is tested with two kinds of $Relative(q_{si})$, namely hypernym sense and hyponym sense. However, the result of LMSR remains almost unchanged. So in the next section, the experiments of the LSM is conducted without hierarchical smoothing.

5.3 Evaluation of Language Sense Model

The results of the LMTR and LSM are compared with different queries and different smoothing methods as shown in Table 4, where a diamond (◇) means the LSM using the Dirichlet smoothing. From the "Improved Map" column, we can see that the LSM outperforms both the traditional language model and vector space model on all queries. From the "Improved Recall" column, we can see that the LSM improved the recall on the medium and long queries as well. The 11-point precision/recall curves for the LSM

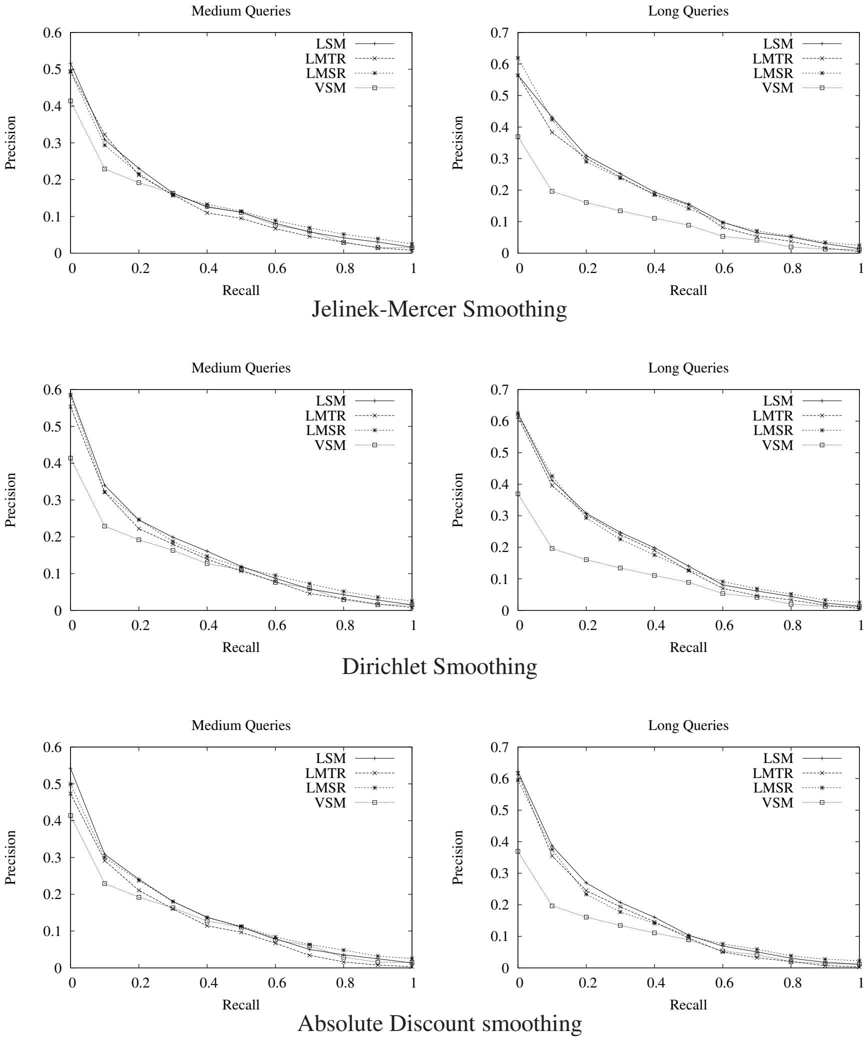

Fig. 3. 11-point precision/recall curves for the LSM, LMTR, LMSR and VSM on Medium and Long Queries

using the Jelinek-Mercer, Dirichlet and Absolute Discount smoothing are shown in Figure 3. In each figure, the four curves from the up-right to bottom-left are LSM, LMTR, LMSR and VSM respectively. To understand whether these improvements are statistically significant, we performed t-tests on MAP. The p-values are shown in the "Sign" column of Table 4 where an asterisk (*) means significant improvement(< 0.05). From the result, we can see that the LSM improves significantly on both medium and long queries, however, not significantly on short queries. It's reasonable because that:

– There are less nouns and verbs to be disambiguated for short queries(see Table 2).
– It's much harder to disambiguate the short queries because of the sparse context.

6 Conclusion and Future Work

In the work, we implement a WSD system which is designed for nouns and verbs only. Then the language model on sense representations (LMSR) and language sense model (LSM) are proposed. The LSM integrated the fine-grained disambiguated senses and terms seamlessly through an EM algorithm. The experiments show that the LSM outperforms both vector space model and traditional language models significantly on both medium and long queries (7.53%-16.90%) with various smoothing methods. From this study, we can also empirically draw that the fine-grained senses will help the information retrieval if they are properly utilized.

In the future, we will study the hierarchical smoothing using more WordNet relations. In addition, we will further evaluate the LSM on more corpus and study how the accuracy of WSD affects the LSM.

References

1. Voorhees, E.M.: Using wordnet to disambiguate word senses for text retrieval. In Korfhage, R., Rasmussen, E.M., Willett, P., eds.: Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993, ACM (1993) 171–180
2. Wallis, P.: Information retrieval based on paraphrase. (1993)
3. Sussna, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In: CIKM '93: Proceedings of the second international conference on Information and knowledge management, ACM Press (1993) 67–74
4. Gonzalo, J., Verdejo, F., Chugur, I., Cigarran, J.: Indexing with WordNet synsets can improve text retrieval. In: Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP, Montreal, Canada (1998) 38–44
5. Sanderson, M.: Word sense disambiguation and information retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Dublin, Ireland (1994) 142–151
6. Krovetz, R.: Viewing Morphology as an Inference Process,. In: Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1993) 191–203
7. Kim, S.B., Seo, H.C., Rim, H.C.: Information retrieval using word senses: root sense tagging approach. In: SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval, ACM Press (2004) 258–265

8. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Research and Development in Information Retrieval. (1998) 275–281
9. Sanderson, M.: Retrieval with good sense. Information Retrieval **2** (2000) 47–67
10. Stokoe, C., Oakes, M.P., Tait, T.: Word sense disambiguation in information retrieval revisited. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Text representation (2003) 159–166
11. Rosenfeld, R.: Two decades of statistical language modeling: Where do we go from here (2000)
12. Song, F., Croft, W.B.: A general language model for information retrieval. In: Proceedings of the eighth international conference on Information and knowledge management. (1999) 316 – 321
13. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems **22** (2004) 179–214
14. Kurland, O., Lee, L.: Corpus structure, language models, and ad hoc information. In: Proceedings of the 27th International ACM SIGIR conference conference. (2004) 194–201
15. Xu, J., Croft, W.: Cluster-based retrieval using language models. In: Proceedings of the 27th International ACM SIGIR conference conference. (2004)
16. Srikanth, M., Srihari, R.K.: Exploiting syntactic structure of queries in a language modeling approach to ir. In: Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana,USA, ACM (2003) 476–483
17. Gao, J., Nie, J.Y., Wu, G., Cao, G.: Dependence language model for information retrieval. In: Proceedings of the 27th annual international conference on Research and development in information retrieval. (2004)
18. Gao, G., Nie, J.Y., Bai, J.: Integrating word relationships into language models. In: Proceedings of 17th ACM SIGIR conference. (2005) 298–305
19. Mihalcea, R.F., Moldovan, D.I.: A highly accurate bootstrapping algorithm for word sense disambiguation. International Journal on Artificial Intelligence Tools **10** (2001) 5–21
20. Liu, S., Liu, F., Yu, C., Meng, W.: An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2004) 266–272
21. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In Joshi, A., Palmer, M., eds.: Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, San Francisco, Morgan Kaufmann Publishers (1996) 310–318