Semantic v.s. Positions: Utilizing Balanced Proximity in Language Model Smoothing for Information Retrieval*

Rui Yan[†,♮], **Han Jiang**[†,♮], **Mirella Lapata**[‡], **Shou-De Lin**[⋆], **Xueqiang Lv**[◇], **and Xiaoming Li**[†]
[♮]School of Electronics Engineering and Computer Science, Peking University
[⋆]Dept. of Computer Science and Information Engineering, National Taiwan University
[‡]Institute for Language, Cognition and Computation, University of Edinburgh
[◇]Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, BISTU
{r.yan, billybob, lxq, lxm}@pku.edu.cn, mlap@inf.ed.ac.uk, sdlin@csie.ntu.edu.tw

Abstract

Work on information retrieval has shown that language model smoothing leads to more accurate estimation of document models and hence is crucial for achieving good retrieval performance. Several smoothing methods have been proposed in the literature, using either semantic or positional information. In this paper, we propose a unified proximity-based framework to smooth language models, leveraging semantic and positional information simultaneously in combination. The key idea is to project terms to positions where they originally do not exist (i.e., zero count), which is actually a word count propagation process. We achieve this projection through two proximity-based density functions indicating semantic association and positional adjacency. We balance the effects of *semantic* and *positional* smoothing, and score a document based on the smoothed language model. Experiments on four standard TREC test collections show that our smoothing model is effective for information retrieval and generally performs better than the state of the art.

1 Introduction

Recently, statistical language models have attracted much attention in the information retrieval community due to their solid theoretical background as well as their success in a variety of retrieval tasks (Ponte and Croft, 1998; Zhai and Lafferty, 2001b). Queries and documents are assumed to be sampled from hidden generative models, and the similarity between a document and a query is then calculated through the similarity between their underlying language models. Clearly, good retrieval performance relies on the accurate estimation of the query and document models. As queries are generally too short (Zhai and Lafferty, 2001a), the entire retrieval problem is essentially reduced to the problem of estimating a document language model (Lavrenko and Croft, 2001; Liu and Croft, 2004).

Larger observed data generally allow people to establish a more accurate statistical model. Unfortunately, in retrieval, we often have to estimate a model based on a small sample of data (e.g., a single document or only a few documents). Therefore, given limited data sampling, a language model estimation sometimes encounters with the zero count problem: the maximum likelihood estimator would give unseen terms a zero probability, which is not reliable because a larger sample of the data would likely contain the term. Language model smoothing is proposed to address the problem, and has been demonstrated to affect retrieval performance significantly (Zhai and Lafferty, 2001b).

To this end, the quality of retrieval tasks heavily relies on proper smoothing of the document language model. Although much work on language model smoothing has been investigated, two related retrieval heuristics remain to be further explored: 1) *intra-document* smoothing, a propagation of word count to positions where the term does not exist, within the local document; 2) *inter-document* smoothing, a projection of non-existence terms from the entire collection globally. Both heuristics are implemented in this paper.

As the key idea is to propagate term counts via intra-document and inter-document projection to positions where they originally do not exist, we have two ways of projection: we propose a unified proximity-based framework to smooth language models, formulating semantic and position information simultaneously into a single objective function with balance. Intuitively, a smoothed language model should enhance the coherence between terms with large semantic association, and

*This work was partially done when Rui Yan was an intern in Intel Center, National Taiwan University
†Indicates equal contributions

International Joint Conference on Natural Language Processing, pages 507–515,
Nagoya, Japan, 14-18 October 2013.

analogously for those positional adjacent terms. In other words, the terms that are close to each other (either semantically related or positionally adjacent) should have similar (smoothed) language models; the closer they are, the more similar their smoothed language models are. The smoothing method is based on two density functions of propagated counts of words. Our proposed framework can combine both semantic and positional proximity for intra-/inter-document smoothing naturally, which has not been addressed in the previous works. To the best of our knowledge, we are the first to balance the effect of these proximities for both intra-/inter-document smoothing.

Another main technical challenge lies in how to define the propagation functions of semantic projection and positional adjacency in order to estimate the language model accordingly. As the adjacency function has been carefully explored in (Lv and Zhai, 2009), we mainly focus on proposing and evaluating several different semantic association functions for term propagation. In these density functions, "close-by" terms would receive more propagated counts than "far-away" terms, which captures the proximity heuristics.

We evaluate the retrieval performance using several standard TREC test collections. Experimental results show that our proposed proximity-based smoothing consistently outperforms the baseline smoothing methods, indicating the effectiveness of our approach. The results show that the derived smoothing method can improve over the baseline position-based smoothing method significantly, and either outperform or perform comparably to the corresponding state-of-art semantic proximity-based smoothing method.

The rest of the paper is organized as follows. We start by reviewing previous works. Then we introduce the balanced language model smoothing, based on semantics and positions separately. We describe the experiments and evaluation in the next section and finally draw the conclusions.

2 Related Work

Language modeling approaches have recently enjoyed much attention for many different tasks ever since the pioneering work applying on information retrieval (Ponte and Croft, 1998). In the past decade, many variants of language models have been proposed, mostly focusing on improving the estimation of query language models (Zhai and Lafferty, 2001a; Lavrenko and Croft, 2001) and document language models (Liu and Croft, 2004; Tao et al., 2006). These methods boil down to retrieval functions that implement retrieval heuristics similar to those implemented in a traditional model, such as TF-IDF weighting and document length normalization (Zhai and Lafferty, 2001b). Yet with sound statistical foundation, language models make it easier to optimize parameters and often outperform traditional retrieval models (Song and Croft, 1999).

Due to the importance of smoothing, many approaches have been proposed and tested. To smooth a document language model, most early smoothing methods relied on using a background language model, which is typically estimated based on the whole document collection (Ponte and Croft, 1998; Zhai and Lafferty, 2001b; Miller et al., 1999). In contrast to the simple strategy which smoothes all documents with the same background, recently corpus structures have been exploited for more accurate smoothing. The basic idea is to smooth a document language model with the documents similar to the document under consideration through clustering (Liu and Croft, 2004; Xu and Croft, 1999; Mei et al., 2008), document expansion (Kurland and Lee, 2004; Tao et al., 2006), or relevance propagation (Kurland and Lee, 2010; Kurland and Lee, 2006; Qin et al., 2005). All these methods are based on document-level semantics similarity to offer "customized" smoothing for each individual document.

Besides semantics, positional heuristics for retrieval have been examined in (Keen, 1992; Tao and Zhai, 2007; Liu and Croft, 2002; Büttcher et al., 2006). Positional language models are proposed to examine the positional proximity in (Lv and Zhai, 2009; Zhao and Yun, 2009). In their work, the key idea is to define a language model for each position within a document, and score it based on the language models on all its positions: hence the effect of positional adjacency is revealed, while semantic information is hardly incorporated.

There is a study in (Karimzadehgan and Zhai, 2010) which smooths language model by term translation model with backgrounds, while we operate term-to-term association on every term position, which is actually a new granularity. Besides, our method takes both semantic and positional information into account, and formulates the two intrinsically different proximity-based heuristics into a unified term-level smoothing framework. To the best of our knowledge, this is the first approach that achieves the combined smoothing.

3 Proximity Based Language Smoothing

We propose a term-level proximity based smoothing approach in the positional language model framework. Each word propagates the evidence of its occurrence to all other positions in the document based on semantic and/or positional projection via density functions. To capture the proximity heuristics, we assign "close-by" words with higher propagated counts than those "far away" from the current word. In other words, most propagated counts come from "nearby" words. Here, *close* and *far* could either be semantic or positional. Each position receives propagated counts of words from an intra-document or an inter-document vocabulary set. All positions have a full vocabulary with different term distributions: each word has a certain non-zero probability to occur in each of the positions, as if all words had appeared in any position with a variety of discounted counts, shown in Figure 1.

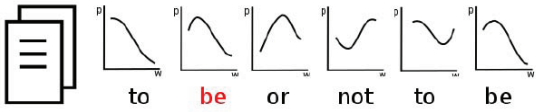


Figure 1: Illustration of different term distributions on different positions for the short document of *"To be or not to be"*. x-axis denotes all terms in vocabulary, while y-axis indicates the term occurrence probability.

3.1 Semantic Proximity based Propagation

The idea for semantic projection is that if a word w occurs at position i, we would like to assume that the highly associated words have also occurred here, with a discounted count. The larger the semantic association is, the larger the propagated count will be. Generally, each propagated count has a value less than 1, which is estimated as the count of w at position i.

Let $d = (w_1, \ldots, w_i, \ldots, w_j, \ldots, w_{|d|})$ be a document, where 1, i, j, and $|d|$ are absolute positions of the corresponding terms in the document, and obviously $|d|$ is the length of the document.

$c(w, i, d)$: the original count of term w at position i in document d before smoothing. If w occurs at position i, $c(w, i, d)$ is 1, otherwise 0. Similarly, $c(w, d)$ is the term count in d and $c(w)$ is the term count within the collection.

$\phi(w_i, w)$: the propagated count of w to position i based on the existence of w_i. Intuitively, $\phi(w_i, w)$ serves as a discounting factor measured by the semantic association between the term w_i and w.

$c'(w, i, d)$: the total propagated count of term w at position i from its occurrences in all the positions in the document d, i.e., $c'(w, i, d) = \sum_{j=1}^{|V|} c(w, j, d)\phi(w_i, w) = c(w, d)\phi(w_i, w)$. Even if $c(w, i, d)$ is 0, $c'(w, i, d)$ may be greater than 0.

Note that the **semantic association** function $\phi(.)$ here is not the same as "similarity". Generally, association denotes the association between two terms based on the broader background, e.g., co-occurrence or mutual information, etc. Clearly, a major technical challenge for semantic based smoothing lies in a proper model to define the association function. We present here 4 representative association calculations.

Co-occurrence Likelihood. Given the term w_i at position i, we calculate the co-occurrence probability for the word w from other positions using:

$$p(w|w_i) = \frac{\#c(w, w_i)}{\#c(w_i)} \tag{1}$$

$\#c(w, w_i)$ is the times of co-occurrence for these two terms. Generally, we need to predefine a sliding window to measure this co-occurrence count, and hence we count $\#c(w, w_i)$ within the same sentence out of the whole collection. $\#c(w_i)$ is the term frequency in the document collection. $p(w|w_i)$ denotes the occurrence probability of w when w_i occurs.

Apparently, this definition is asymmetric because $p(w|w_i) \neq p(w_i|w)$. When calculate the propagated counts for w, it is more reasonable to measure the probability given the existence of w_i. Especially when w_i is a low-frequency term, we will find the most likely terms with high co-occurrence probability. The semantic association by co-occurrence likelihood is $\phi_{cl}(w_i, w) = p(w|w_i)$.

Mutual Information. In Information Theory, the mutual information of two random variables is a quantity that measures their mutual dependence, which in our case, is the dependence of co-occurrence probability. The mutual information between the two terms w and w_i can be represented as:

$$\text{MI}(w_i, w) = \log \frac{p(w, w_i)}{p(w)p(w_i)} \tag{2}$$

where

$$p(w_i, w) = p(w|w_i)p(w_i) \tag{3}$$

$p(w|w_i)$ is defined in Equation (1), and $p(w) =$

$\frac{\#c(w)}{\sum_{w' \in V} \#c(w')}$. Equation (2) can be rewritten as:

$$\text{MI}(w_i, w) = \log\left(\frac{\#c(w, w_i)}{\#c(w)\#c(w_i)} \sum_{w' \in V} \#c(w')\right) \quad (4)$$

Generally, a larger value of mutual information between terms indicates larger association while low value or negative value indicates independency. Although low mutual information is proved to be less dependent, high mutual information does not necessarily guarantee high association, especially for low-frequency terms. Therefore, we apply the Refined Mutual Information (RMI) as an improvement (Manning and Schütze, 1999).

$$\text{RMI}(w_i, w) = \begin{cases} \#c(w, w_i)\text{MI}(w, w_i) \\ 0 \quad (\text{if MI}(w, w_i) < 0) \end{cases} \quad (5)$$

Finally, we normalize RMI into [0, 1] by using RMI_{max}, the maximum value of RMI, as the semantic association by mutual information:

$$\phi_{mi}(w_i, w) = \frac{\text{RMI}(w, w_i)}{\text{RMI}_{\text{max}}} \quad (6)$$

Thesaurus-Based Correlation. A word thesaurus represents the semantic associations of terms, which is often formed into a tree with synonyms, hyponyms and hypernyms modeled by "parent-to-child" relationships, e.g., WordNet[1] or Wikipedia[2]. We illustrate part of WordNet as follows in Figure 2:

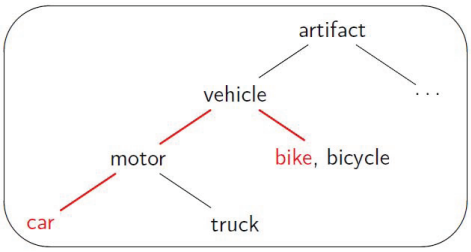


Figure 2: Hierarchical structure of WordNet: red lines imply a possible path between *car* and *bike*.

There could be many paths from one term to the other, and we define the shortest path as the distance between two terms, denoted as $dist(w_i, w)$. Intuitively, the shorter distance is, the larger semantic association is expected. Hence, we utilize a decreasing sigmoid function to model the semantic association based on thesaurus, denoted as ϕ_{tc}:

$$\phi_{tc}(w_i, w) = \frac{1}{1 + \text{e}^{(dist(w_i, w))}} \quad (7)$$

[1] http://wordnet.princeton.edu
[2] http://wikipedia.org

Topic Distribution. "Topics" have long been investigated as the significant latent aspects for linguistic analysis (Hofmann, 2001; Landauer et al., 1998). The utilization of topic models provides a new horizon to investigate the latent correlations between terms and documents. We apply the unsupervised Latent Dirichlet Allocation (Blei et al., 2003) to discover topics[3]. We obtain the probability distribution over topics assigned to a term w, i.e., $p(w|z)$. The inferred topic representation is the probabilities of terms belonging to the topic z, which is

$$z = \{p(w_1|z), p(w_2|z), \ldots, p(w_i|z)\}$$

We empirically train a k-topic model (k=100) and invert the topic-term representation in Table 1, where each w is represented as a topic vector $\vec{w}$. The semantic association based on topic distribution $\phi_{td}(w_i, w)$ between w_i and w is measured by the cosine similarity on topic vector $\vec{w}_i$ and $\vec{w}$.

$$\phi_{td}(w_i, w) = \frac{\vec{w}_i \cdot \vec{w}}{||\vec{w}_i|| ||\vec{w}||} \quad (8)$$

Table 1: Inverted *topic-term* vector representation.

$\vec{w}_1$	$p(w_1\|z_1)$	$p(w_1\|z_2)$	...	$p(w_1\|z_k)$
$\vec{w}_2$	$p(w_2\|z_1)$	$p(w_2\|z_2)$	...	$p(w_2\|z_k)$
⋮	⋮	⋮	⋮	⋮
$\vec{w}_{\|V\|}$	$p(w_V\|z_1)$	$p(w_V\|z_2)$	...	$p(w_V\|z_k)$

3.2 Intra-/Inter-Document Smoothing

For every position i to estimate the language model, we can project a term from other positions within the document through the defined semantic association functions, namely *intra-document* smoothing. We can also project all terms from the whole vocabulary set to position i via $\phi(.)$, which is actually an *inter-document* smoothing effect from the global collection and hence solve the zero probability problem.

Before smoothing, the original word count distribution for position i in document d is $\mathcal{D}(i, d)$, with only $c(w_i, i, d)$=1 while all other items are 0.

$$\mathcal{D}(i, d) = \left[\underbrace{[c(w_1, i, d), \ldots, c(w_i, i, d), \ldots, c(w_{|V_d|}, i, d)]}_{V_d}, \underbrace{0, \ldots, 0}_{V \setminus V_d}\right]$$

After the semantic based intra-document smoothing, the word count distribution becomes:

$$\mathcal{D}_s(i, d) = \left[\underbrace{[c'(w_1, i, d), \ldots, c'(w_i, i, d), \ldots, c'(w_{|V_d|}, i, d)]}_{V_d}, \underbrace{0, \ldots, 0}_{V \setminus V_d}\right]$$

[3] We use Stanford TMT (http://nlp.stanford.edu/software/tmt/), with default parameter settings.

Imagine the whole collection as a long *virtual document*, the terms outside the document vocabulary of V_d could also be smoothed by inter-document smoothing, i.e., $c'(w,i) = c(w)\phi(w_i,w)$. To control the impact of out-of-document vocabulary, we add a parameter $\mu \in [0,+\infty)$ here:

$$\mathcal{D}_s(i,d) = \Big[\underbrace{[c'(w_1,i,d),\ldots,c'(w_{|V_d|},i,d)]}_{V_d}, \underbrace{\mu c'(w_j,i),\ldots,\mu c'(w_{|V|},i)}_{V\setminus V_d}\Big]$$

3.3 Positional Proximity based Propagation

Analogously, for the positional-based smoothing, the smoothed count by positional proximity is $c''(w,i,d)=\sum_{j=1}^{|V|} c(w,j,d)\psi(i,j)$. We apply the best positional proximity based density function of Gaussian projection $\psi(i,j)$ in (Lv and Zhai, 2009). σ is a fixed parameter here.

$$\psi(i,j) = \exp[\frac{-\Delta(i,j)^2}{2\sigma^2}] \tag{9}$$

Analogously to the semantic smoothing, we also include the intra-/inter-document smoothing in the positional count propagation. It is natural to measure the distance offset between two terms within the same document. To measure the position distance between terms from different documents, we define $\Delta(i,j) = +\infty$ when the term w at a certain position j is not from the document which contains w_i, i.e.,

$$\Delta(i,j) = \begin{cases} |i-j| & (w_j \in d) \\ +\infty & (w_j \notin d) \end{cases} \tag{10}$$

In this way, the projection value of $\psi(i,j)$ is calculated to be 0 when $w_j \notin d$. Actually the definition is rather flexible, the value of projection for terms from different documents is easy to adjust to be non-zero when Equation (10) is changed.

The word count distribution $\mathcal{D}_p(i|d)$ is as follows after positional proximity based smoothing:

$$\mathcal{D}_p(i,d) = \Big[\underbrace{[c''(w_1,i,d),\ldots,c''(w_i,i,d),\ldots,c''(w_{|V_d|},i,d)]}_{V_d}, \underbrace{0,\ldots,0}_{V\setminus V_d}\Big]$$

3.4 Balanced Proximity Combination

We can estimate a language model for the position i based on the propagated counts reaching the position. Since we have two smoothed language distributions, i.e., $\mathcal{D}_s(i|d)$ and $\mathcal{D}_p(i|d)$, with uniform representation, we can combine both smoothing strategies with balance by distribution function superposition, illustrated in Figure 3.

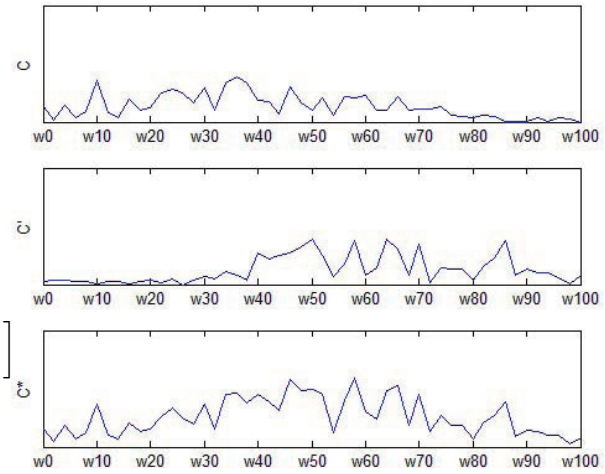


Figure 3: Linear interpolation of two smoothed language models. The upper two word distributions are overlaid into one.

Based on the term propagation, we have a term frequency vector $<w_1, w_2, \ldots, w_{|V_d|}, \ldots, w_V>$ at position i, forming a virtual document d_i. λ is to control the relative contributions from semantic proximity based smoothing and positional proximity based smoothing, formulated as:

$$\begin{aligned}\hat{c}(w,i,d) =& \lambda c'(w,i,d) + (1-\lambda)c''(w,i,d) \\ =& \lambda \sum_{j=1}^{|V|} c(w,j,d)\phi(w_i,w) + \\ & (1-\lambda)\sum_{j=1}^{|V|} c(w,j,d)\psi(i,j)\end{aligned} \tag{11}$$

$\phi(w_i,w)$ is to measure the semantic association between w and the word w_i from position i, and $\psi(i,j)$ is the distance discount factor.

Thus the language model of this virtual document can be estimated as:

$$p(w|i,d) = \frac{\hat{c}(w,i,d)}{\sum_{w'\in V}\hat{c}(w',i,d)} \tag{12}$$

where V is the vocabulary set. $\sum_{w'\in V}\hat{c}(w',i,d)$ is actually the length of the virtual document. $p(w|i,d)$ is the language model at position i. Thus given a query q, we can adopt the KL-divergence retrieval model (Lafferty and Zhai, 2001) to score each language model at every position as follows:

$$Score(q,d,i) = -\sum_{w\in V} p(w|q)\log\frac{p(w|q)}{p(w|i,d)} \tag{13}$$

$p(w|q)$ is a query language model. We apply the 1) *best* scoring and 2) *average* scoring of all positions in the document as the retrieval ranking strategy (Lv and Zhai, 2009).

4 Experiments

4.1 Dataset and Evaluation

In this section, we evaluate the effectiveness of our smoothing strategies empirically. We use four representative TREC data sets: AP (Associated Press news 1988-90), LA (LA Times), WSJ (Wall Street Journal 1987-92) and TREC8 (Disk 4 & 5, the ad hoc data used in TREC8). They represent different sizes and genres, with the same source, queries, and preprocessing procedure as in (Tao et al., 2006; Lv and Zhai, 2009). Table 2 shows the basic statistics of these datasets in detail. We used the title field of a TREC topic description to simulate short keyword queries in our experiments.

Table 2: Detailed basic information of 4 datasets.

	AP	LA	WSJ	TREC8
#doc	242,918	131,896	173,252	528,155
avg(dl)	442.4	492.5	388.7	468.3
qry id	51-150	301-400	51-100 151-200	401-450
#qry	100	100	100	50
#t_qrel	21,819	2,350	10,141	4,728
avg(ql)	4.55	2.63	4.68	2.46

#doc/#qry: number of docs/queries; #t_qrel: number of relevant docs; avg(dl)/avg(ql): average length of doc/qry.

In each experiment, we first use the baseline model (KL-divergence) to retrieve 2,000 documents for each query, and then use the smoothing methods (or a baseline method) to re-rank them. The top-ranked 1,000 documents for all runs are compared using P@10 and the Mean Average Precisions (MAP) as the main metric.

$$\text{MAP} = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{|D_r|}\sum_{i=1}^{k} P_i \times rel_i$$

$|Q|$ is the number of queries, $|D_r|$ denotes the total number of relevant documents, P_i is the precision at i-th position for, also called P@i (Manning et al., 2008). rel_i is an indicator function equaling 1 if the item at rank i is a relevant document, 0 otherwise.

4.2 Algorithms for Comparison

We examine the retrieval performance on the standard datasets. The first baseline group is based on the traditional language model. **LM** is the language model without smoothing at all, while **LM+JM** and **LM+Diri** are to smooth the language model with the whole collection as background information, using Jelinek-Mercer (JM) and Dirichlet (Diri) smoothing methods correspondingly (Zhai and Lafferty, 2001b).

We also examine a series of semantic based language smoothing. The most representative semantic smoothing is the Cluster-Based Document Model (**CBDM**) proposed by Liu et al. (2004). We apply the default settings for the method, (e.g., clustering methods, etc). Semantic based methods use semantically similar documents as a smoothing corpus for a particular document: CBDM clusters documents and smooths a document with the cluster where that document belongs to. However, this method is only based on document-level semantic similarity rather than term-level semantic association.

We also include Positional Language Model (**PLM**) proposed by Lv et al. (2009), which is the state-of-art positional proximity based language smoothing. PLM mainly utilizes positional information while no semantic association is considered. We implemented the best reported PLM kernel with Dirichlet smoothing from the collection for comparison.

Finally we include our proposed Balanced Proximity-based Model, denoted as **BPM**, which formulates semantic proximity and positional proximity into a unified language smoothing framework, with flexible intra-document smoothing and inter-document smoothing. In all, we have 7 methods to compare their performance.

4.3 Overall Performance Comparison

In this section, we compare BPM smoothing with several previously proposed methods, using the Dirichlet smoothing prior which performs best as mentioned in these works. The prior parameter is set at 1000 for all methods to rule out any potential influence of Dirichlet smoothing (Liu and Croft, 2004; Tao et al., 2006). For fairness, we conduct the same pre-processing to all methods. The parameter is chosen by 10-fold cross validation.

For baselines, we use the source code from the original author, and report the results we get. The advantage of CBDM, PLM (and BPM) over the simplest language smoothing with Dirichlet and Jelinek-Mercer smoothing has long been proved. We hence focus on the meaningful comparison between the sophisticated smoothing techniques. Note that under the real scenario, as we could not always predefine which kernel would perform best on a particular dataset, for fairness, we take the average performance of all semantic association kernels as the results of BPM, and the parameters are chosen using 10-fold cross validation described in Section 4.4.2. Tables 3 and 4 show that our model outperforms PLM in MAP and P@10 values on four data sets. The improvement presumably comes from the combination of both semantic and positional proximity based smooth-

MAP	LM	LM+JM	LM+Diri	CBDM	PLM	BPM
AP	0.169133	0.179245	0.180625	0.204361	0.204216	**0.207428*****
LA	0.204195	0.222077	0.219500	**0.240332**	0.221190	0.231593
WSJ	0.206986	0.220919	0.221066	0.253834	0.269038	**0.277115****
TREC8	0.181040	0.214923	0.209676	0.219018	0.240894	**0.248852**
P@10	LM	LM+JM	LM+Diri	CBDM	PLM	BPM
AP	0.402020	0.403030	0.403030	**0.432323**	0.418501	0.400000
LA	0.251020	0.256122	0.245918	0.288776	0.278571	**0.289913***
WSJ	0.365142	0.369204	0.379826	0.435036	0.423628	**0.446802***
TREC8	0.360508	0.368204	0.358496	**0.442242**	0.425900	0.438028
Time (in sec)	LM	LM+JM	LM+Diri	CBDM	PLM	BPM
PerQuery	0.151803	0.174667	0.180906	337.08198	0.683829	0.918593

Table 3: Overall performance comparison on MAP and P10 results among all methods. *, **, *** indicate that we accept the improvement hypothesis of BPM over the best rival baseline by Wilcoxon test at a significance level of 0.1, 0.05, 0.01 respectively. Efficiency is measured in seconds.

ing; intuitively, the lower bound of BPM is the performance of PLM by tuning the combination parameter λ fixed at 1, which is actually a special case for BPM. It is interesting to find that CBDM based on semantic smoothing performs well in some datasets. We further examine into the datasets of LA and TREC8: in these sets, the semantic proximity weights more than positional proximity, i.e., a smaller λ in Figure 4. As CBDM conducts a more principled way of exploiting semantic smoothing by clustering structures, it should not be too surprising for its performance on datasets which emphasize semantic proximity.

Efficiency. The LM group is naturally faster without sophisticated calculations. BPM is a little slower than PLM but with consistent better performance. CBDM shows the lowest efficiency due to mass calculations of similarity for clustering.

4.4 Strategy Analysis

Generally speaking, strategies can be sorted into two categories: component selection and parameter tuning. Each time, we tune one strategy while the other one remains fixed.

4.4.1 Component Selection

There is one substitutive component of designing the semantic propagation function, where the term association can be calculated by co-occurrence likelihood ϕ_{cl}, mutual information ϕ_{mi}, thesaurus-based correlation ϕ_{tc} and topic distribution ϕ_{td}. We examine the performance of different functions to calculate the semantic association and the results are listed in Table 4 and 5.

From the tables above, we can see that most of the semantic association functions have slightly different performance, indicating that these four measurements are all able to capture the semantic proximity based association among terms. Among all semantic proximity functions, the co-occurrence likelihood ϕ_{cl} performs best in most cases, which means it is reasonable and most natural to smooth the zero count of terms if the co-occurred terms appear.

MAP	ϕ_{cl}	ϕ_{mi}	ϕ_{tc}	ϕ_{td}
AP	**0.208971**	0.207159	0.206713	0.206868
LA	**0.231850**	0.231557	0.231482	0.231483
WSJ	0.276261	**0.278372**	0.276829	0.276999
TREC8	0.242348	**0.251085**	0.250977	0.250996

Table 4: MAP of different semantic associations.

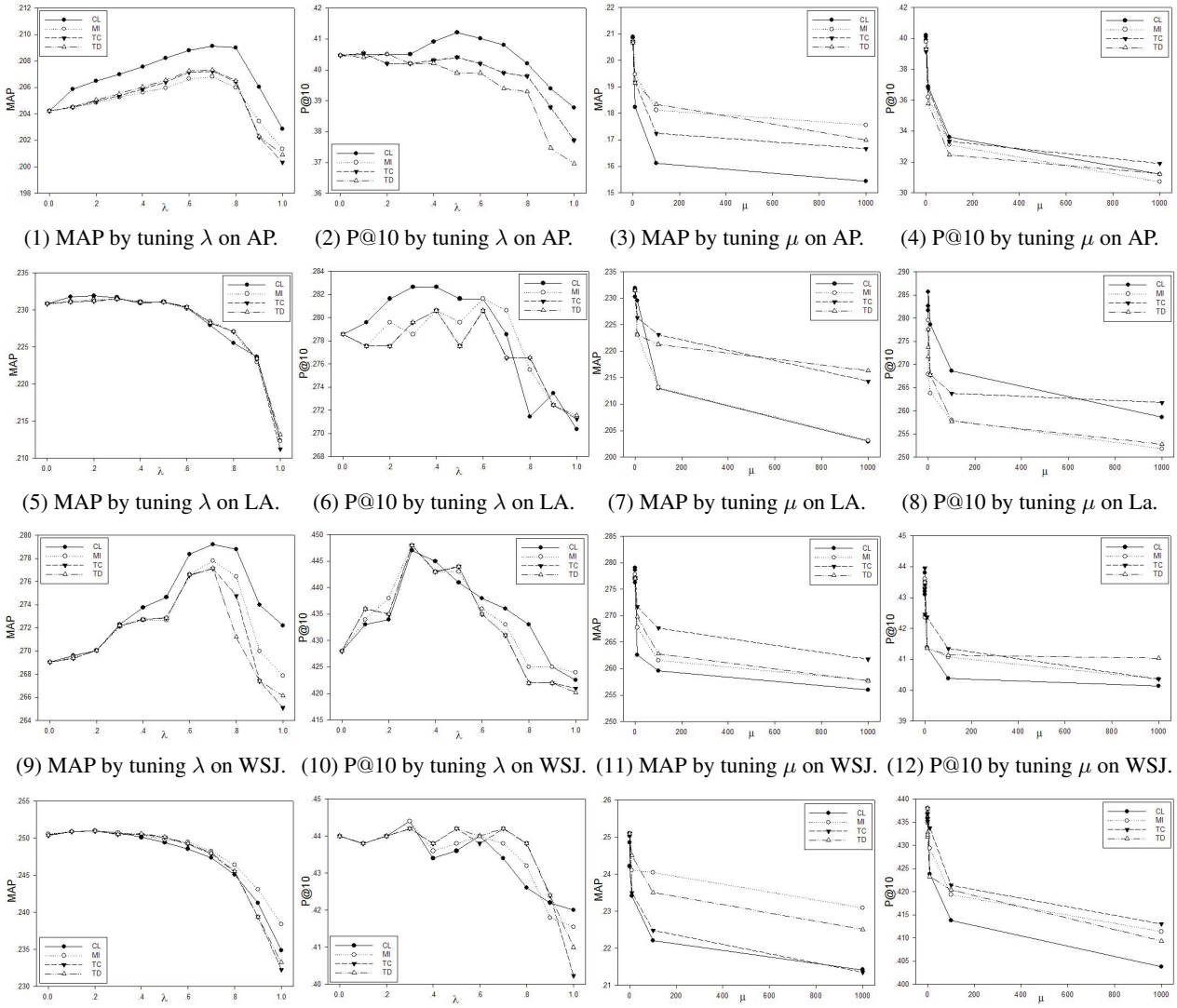
P@10	ϕ_{cl}	ϕ_{mi}	ϕ_{tc}	ϕ_{td}
AP	**0.409091**	0.392929	0.398990	0.398991
LA	**0.294898**	0.285571	0.288592	0.290592
WSJ	**0.447102**	0.436101	0.446986	0.446019
TREC8	0.438008	**0.438103**	0.437998	0.438002

Table 5: P@10 of different semantic associations.

4.4.2 Parameter Settings

There are two free parameters to tune, i.e., λ and μ. λ is to balance the relative contributions from semantic proximity and positional proximity, while μ is to control the weight of inter-document smoothing from the whole collection. Keeping one parameter fixed, we vary the other one to examine the changes of its performance based on all datasets. For each of the 4 datasets, we divide the set and use the 10-fold cross validation to train parameters for testings. We illustrate the performance of parameter sensitivity by tuning λ and μ based on all semantic association kernels, as shown in Figure 4.

To control the tradeoff between semantic and positional proximity combination, we gradually



(1) MAP by tuning λ on AP. (2) P@10 by tuning λ on AP. (3) MAP by tuning μ on AP. (4) P@10 by tuning μ on AP.

(5) MAP by tuning λ on LA. (6) P@10 by tuning λ on LA. (7) MAP by tuning μ on LA. (8) P@10 by tuning μ on La.

(9) MAP by tuning λ on WSJ. (10) P@10 by tuning λ on WSJ. (11) MAP by tuning μ on WSJ. (12) P@10 by tuning μ on WSJ.

(13) MAP by tuning λ on TREC(14) P@10 by tuning λ on TREC(15) MAP by tuning μ on TREC(16) P@10 by tuning μ on TREC.

Figure 4: Examine the sensitivity of λ and μ by all semantic association functions on all datasets.

change λ from 0 to 1 at the step of 0.1 to examine the effect in Figure 4. The combination of both proximity outperforms the performance in isolation (λ = 1 or 0). An interesting observation is that due to the instinct difference of used queries and datasets, the optimal λ varies from one set to another: for AP and WSJ, a larger λ is needed and for LA and TREC8, a small λ is desired perhaps due to the semantic association is more biased for these datasets/corresponding queries: in general, the combination is a better strategy.

We then examine the impact of out-of-document vocabulary controlled by μ in Figure 4. Although the performance varies on different datasets as well, for MAP, the performance is generally downward when μ grows larger, and for P@10, the performance achieves best when μ is relatively small (μ=0.1 or 0.01), which indicates the impact of inter-document smoothing should not be excessively over introduced.

5 Conclusions

In this paper, we combined both semantic and positional proximity heuristics to improve the effect of language model smoothing, which has not been addressed before. We proposed and studied four different semantic proximity-based propagation functions as well as the positional proximity density function to estimate the smoothed language model. Experimental results show that BPM outperforms most alternative baselines in terms of MAP and P@10, which indicates the effectiveness of our proposed method.

Besides the effective fusion of semantic and positional proximity ($\lambda \neq 0$), we further investigate the semantic propagation function, and find that co-occurrence likelihood association performs best. In the future, we will incorporate corpus information such as clustering features into the semantic proximity function for better smoothing.

Acknowledgments

This work was supported by "III Innovative and Prospective Technologies Project" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China, and by National Natural Science Foundation of China (Grant No. 61271304) and Key Program of Beijing Municipal Natural Science Foundation (Grant No. KZ201311232037).

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

Stefan Büttcher, Charles L. A. Clarke, and Brad Lushman. 2006. Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proceedings of SIGIR '06*, pages 621–622.

Thomas Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196. 10.1023/A:1007617005950.

Maryam Karimzadehgan and ChengXiang Zhai. 2010. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of SIGIR '10*, pages 323–330.

E.Michael Keen. 1992. Some aspects of proximity searching in text retrieval systems. *Journal of Information Science*, 18(2):89–98.

Oren Kurland and Lillian Lee. 2004. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of SIGIR '04*, pages 194–201.

Oren Kurland and Lillian Lee. 2006. Respect my authority!: Hits without hyperlinks, utilizing cluster-based language models. In *Proceedings of SIGIR '06*, pages 83–90.

Oren Kurland and Lillian Lee. 2010. Pagerank without hyperlinks: Structural reranking using links induced by language models. *ACM Trans. Inf. Syst.*, 28(4):18:1–18:38, November.

John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR '01*, pages 111–119.

T.K. Landauer, P.W. Foltz, and D. Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of SIGIR'01*, pages 120–127.

Xiaoyong Liu and W. Bruce Croft. 2002. Passage retrieval based on language models. In *Proceedings of CIKM '02*, pages 375–382.

Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of SIGIR '04*, pages 186–193.

Yuanhua Lv and ChengXiang Zhai. 2009. Positional language models for information retrieval. In *Proceedings of SIGIR '09*, pages 299–306.

C.D. Manning and H. Schütze. 1999. *Foundations of statistical natural language processing*, volume 999. MIT Press.

C.D. Manning, P. Raghavan, and H. Schutze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

Qiaozhu Mei, Duo Zhang, and ChengXiang Zhai. 2008. A general optimization framework for smoothing language models on graph structures. In *Proceedings of SIGIR '08*, pages 611–618.

David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A hidden markov model information retrieval system. In *Proceedings of SIGIR '99*, pages 214–221.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of SIGIR'98*, pages 275–281.

Tao Qin, Tie-Yan Liu, Xu-Dong Zhang, Zheng Chen, and Wei-Ying Ma. 2005. A study of relevance propagation for web search. In *Proceedings of SIGIR '05*, pages 408–415.

Fei Song and W. Bruce Croft. 1999. A general language model for information retrieval. In *Proceedings of CIKM '99*, pages 316–321.

Tao Tao and ChengXiang Zhai. 2007. An exploration of proximity measures in information retrieval. In *Proceedings of SIGIR '07*, pages 295–302.

Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings HLT-NAACL '06*, pages 407–414.

Jinxi Xu and W. Bruce Croft. 1999. Cluster-based language models for distributed retrieval. In *Proceedings of SIGIR '99*, pages 254–261.

Chengxiang Zhai and John Lafferty. 2001a. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM'01*, pages 403–410.

Chengxiang Zhai and John Lafferty. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'01*, pages 334–342.

Jinglei Zhao and Yeogirl Yun. 2009. A proximity language model for information retrieval. In *Proceedings of SIGIR '09*, pages 291–298.