Context-Sensitive Semantic Smoothing using Semantically Relatable Sequences

Kamaljeet S Verma
Department of Computer Science
Indian Institute of Technology Bombay
kamaljeet_1290@yahoo.com

Pushpak Bhattacharyya
Department of Computer Science
Indian Institute of Technology Bombay
pb@cse.iitb.ac.in

Abstract

We propose a novel approach to context sensitive semantic smoothing by making use of an intermediate, "semantically light" representation for sentences, called *Semantically Relatable Sequences (SRS)*. SRSs of a sentence are tuples of words appearing in the semantic graph of the sentence as linked nodes depicting dependency relations. In contrast to patterns based on consecutive words, SRSs make use of groupings of non-consecutive but semantically related words. Our experiments on TREC AP89 collection show that the mixture model of SRS translation model and Two Stage Language Model (TSLM) of Lafferty and Zhai achieves MAP scores better than the mixture model of MultiWord Expression (MWE) translation model and TSLM. Furthermore, a system, which for each test query selects either the SRS or the MWE mixture model based on better query MAP score, shows significant improvements over the individual mixture models.

1 Introduction

Ponte and Croft [Ponte and Croft, 1998] first proposed the language modeling approach to text retrieval. The simplicity and effectiveness of the approach provided the IR researchers with a new attractive text retrieval framework. The main idea of this approach is to first estimate the document model and then calculate the query generation likelihood according to the estimated model. An important step in the estimation of document models called *Smoothing* is crucial to boost the retrieval performance. Since the query terms may not appear in the document, some reasonable non-zero probability must be assigned to unseen terms and also the probability of seen terms must be adjusted to remove the noise. Document smoothing considers both these cases while estimating the models. Various smoothing techniques have been proposed by IR researchers as in [Berger and Lafferty, 1999][Lafferty and Zhai, 2001][Zhai and Lafferty, 2001][Zhou *et al.*, 2007a]. Initial approaches like the one by Berger and Lafferty [Berger and Lafferty, 1999] were able to incorporate synonym and sense information into the language models. Later on, approaches to incorporate context into the models were proposed in [Zhou *et al.*, 2007a][Zhou *et al.*, 2006].

In this paper, we propose the use of a representation for sentences called *Semantically Relatable Sequences (SRS)* for document smoothing. The approach we have adopted is comparable to the Topic Signature Language Modeling approach proposed by Zhou et al. in [Zhou *et al.*, 2007a]. However, their approach relies on multiword expressions to perform context-sensitive semantic smoothing. But multiwords are limited by the constraint of consecutivity. It has in general been unclear as to how to incorporate context when the related words are far apart in the sentence.

Our chief contribution is suggesting a solution to the problem of context sensitive semantic smoothing by making use of Semantically Relatable Sequences (SRS) that are tuples capturing semantically related, but not necessarily consecutive, words. The roadmap of the paper is as follows. Section 2 surveys literature on context sensitive semantic smoothing. Section 3 defines Semantically Relatable Sequences and elucidates the concept with many examples. Section 4 discusses the SRS based translation model focusing on the methodology of document smoothing using SRS. Section 5 details out the experiments and presents the results. Section 6 discusses the results. Section 7 concludes the paper.

2 Previous Work

Berger and Lafferty [Berger and Lafferty, 1999] proposed a word to word statistical translation model as expressed by Equation (1) below for computing ranking.

$$p(q/d) = \sum_{w} p(q/w) * p(w/d) \tag{1}$$

where $p(q/w)$ is the document word w to query term q translation probability and $p(w/d)$ is the unigram language model. Although this model was able to incorporate synonyms and sense information into the language models, it failed to capture context. Thus for example, the term *case* might get translated to *lawsuit* or *container* with equal probabilities, irrespective of context.

Recently, a context sensitive approach called Topic Signature Language Modeling was proposed by Zhou et al. [Zhou *et al.*, 2007a]. In this approach, a document is decomposed into topic signatures which are then statistically translated to query terms. For general domain, multiword expressions extracted by Xtract [Smadja, 1993] are used as topic signatures. The equation below describes this

$$p(w/d) = \sum_k p(w/t_k) * p(t_k/d) \tag{2}$$

where $p(w/t_k)$ is the topic signature t_k to term w translation probability and $p(t_k/d)$ is the topic signature generation probability given document d. Since, multiword expressions contain contextual information and are mostly unambiguous, the translation probabilities are more specific and the smoothed document models have high accuracy.

Latent topic models such as Probabilistic Latent Semantic Indexing [Hofmann, 1999] are also very similar to the topic signature language models. The major difference lies in the two models' parameter estimation procedures.

Linguistically-motivated representations have been used before as in [Gao *et al.*, 2005] for representing documents and computing relevance scores in a different way.

3 Semantically Relatable Sequences

Words in natural language text can be classified as content words or function words. The former are nouns, adjectives, verbs and adverbs, while the latter are prepositions, conjunctions, articles etc. It has been postulated [Mohanty *et al.*, 2005] that a sentence needs to be broken into sequences of at most three types: *(CW, CW)*, *(CW, FW, CW)* and *(FW, CW)*. *CW* represents a simple content word or a compound concept, *FW* a function word. Based on this, SRSs have been defined in [Mohanty *et al.*, 2005] as follows:

Definition: A semantically relatable sequence (SRS) of a sentence is a group of words in the sentence, not necessarily consecutive, that appear in the semantic graph of the sentence as linked nodes or nodes with speech act labels.

Example-1: *The man bought a new car in June.*
Content Words: *man, bought, new, car, June*
Function words: *the, a, in*
SRSs:

1. {*man, bought*}
2. {*bought, car*}
3. {*bought, in, June*}
4. {*new, car*}
5. {*the, man*}
6. {*a, car*}

Note how the representation uncovers the direct dependencies in the sentence, including the long distance one between *bought* and *June*.

3.1 Capturing clauses and compounds: SCOPE

SRSs can be used to represent different kinds of sentential constituents.

Example-2: *We know that Google acquired the search engine company Oingo in 2003.*
SRSs:

1. {*We, know*}
2. {*know, SCOPE*}
3. *SCOPE:*{*Google, acquired*}
4. *SCOPE:*{*acquired, company*}
5. *SCOPE:*{*acquired, Oingo*}
6. *SCOPE:*{*search, company*}
7. *SCOPE:*{*engine, company*}
8. *SCOPE:*{*acquired, in, 2003*}
9. *SCOPE:*{*the, company*}

The embedded clause *Google acquired the search engine company Oingo in 2003* is expressed under a *SCOPE*. A *SCOPE* provides an umbrella for words occurring in a clause or involved in compounding. The semantic relation between the embedded clause and the words in the main clause is depicted through the SRS {*know, SCOPE*}.

Example-3: *John and Mary went to school.*
SRSs:

1. {*SCOPE, went*}
2. *SCOPE:*{*John, and, Mary*}
3. {*went, to, school*}

The SRS tuple {*John, and, Mary*} represents a compound concept and is marked under *SCOPE*.

3.2 SRS Generation

SRS generation is a complex process. The parse tree of the input sentence is first generated using the Charniak Parser [Charniak, 2000]. Each node of the parse tree is then processed breadth-first. The tag, the head word and the neighbouring word information is used to finally generate the SRSs. Resources like WordNet [Miller, 1994], Oxford Advanced Learner Dictionary [Hornby, 2001], subcategorization database, etc. are used by the SRS generator. A detailed description of the SRS generation algorithm and usage can be found in [Mohanty *et al.*, 2005][Khaitan *et al.*, 2007]. The document models are expanded by statically mapping useful SRSs to query terms.

4 SRS Based Translation Model

Figure 1 shows the high level architecture diagram of the search engine. The *Indexer* takes the raw documents and the SRS documents as input and generates two types of indexes. The *Translation Probability Estimator* takes both the indexes and generates a huge SRS to word translation probability matrix. The *Searcher* module uses the indexes and the translation probability matrix to rank the documents and the *Evaluator* module evaluates the performance of the searcher module.

4.1 Indexing

The *Indexer* module generates two type of indexes: word index and SRS index. We use the open source language modeling toolkit called the Dragon Toolkit [Zhou *et al.*, 2007b] for index generation.

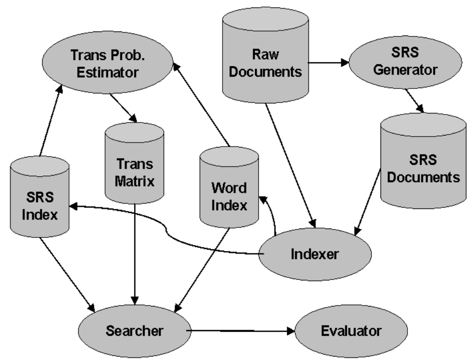

Figure 1: High Level System Architecture

Word Index

Word index similar to the one generated by traditional keyword based search engines is created from the documents. Before creating the index, stop words are removed. A 319 word stop word list compiled by van Rijsbergen [Van Rijsbergen, 1979] is used to identify the stop words. Also, words are stemmed using the Porter Stemmer [Porter, 1997].

SRS Index

The SRS generator module described in section 3.2 is used to generate SRS documents from raw text documents. Once the SRS documents are generated, we generate the SRS index and retain a given SRS_i in the SRS index if it satisfies the following two conditions.

1. SRS_i appears in more than one documents and has frequency 10 or more in the corpus.
2. SRS_i predicts, as described below, atleast 5 other SRS_js in the corpus.

We use the mutual information statistic in (3) to identify SRSs that occur more often than chance, comparing the probability $p(SRS_i, SRS_j)$ of observing the SRSs SRS_i and SRS_j together with the probability of observing SRS_i and SRS_j independently ($p(SRS_i)$ and $p(SRS_j)$ respectively).

$$\frac{p(SRS_i SRS_j)}{p(SRS_i)p(SRS_j)} \tag{3}$$

If this mutual information value exceeds a threshold, we assume that $SRSi$ predicts $SRSj$. In our experiments, we used the threshold of 150 to ensure that good SRSs are retained in the index.

4.2 SRS Translation Model

We estimate the SRS translation model θ, like [Zhou *et al.*, 2007a] estimate their MWE translation model. Specifically, we use the EM algorithm, which starts with an initial guess of the parameter values, and then iteratively improves the estimate by increasing the likelihood until the likelihood converges. The EM update formulas are:

$$p^{(n)}(w) = \frac{(1-\eta)p^{(n)}(w/\theta)}{(1-\eta)p^{(n)}(w/\theta) + \eta p^{(n)}(w/C)} \tag{4}$$

$$p^{(n+1)}(w/\theta) = \frac{\sum_{j=1}^{m} c(w;d_j)p^{(n)}(w)}{\sum_i \sum_{j=1}^{m} c(w_i;d_j)p^{(n)}(w_i)} \tag{5}$$

The *purify* effect achieved by traditional feedback methods closely resembles this estimation method. Table 1 shows top 10 related words corresponding to some sample SRSs which are significant (those crossing a threshold).

4.3 SRS Based Document Smoothing

Once the word and SRS index are generated and SRS to term translation probabilities are estimated, we use them to perform document smoothing. The word translation language model in [Berger and Lafferty, 1999] decomposes a document into words and then statistically maps those words to query terms. The topic signature language model with multiword expressions as topic signatures of [Zhou *et al.*, 2007a] decomposes a document into multiword expressions and maps the multiword expressions to query terms. On similar lines, our SRS based translation language model decomposes a document into SRSs and then statistically maps the SRSs to query terms. The following formula is used to obtain a document model:

$$p(w/d) = \sum_k p(w/SRS_k) * p(SRS_k/d) \tag{6}$$

The probability $p(SRS_k/d)$ of generating SRS_k by the document d can be easily computed by the maximum likelihood estimate formula mentioned below:

$$p(SRS_k/d) = \frac{c(SRS_k, d)}{\sum_i c(SRS_i, d)} \tag{7}$$

where $c(SRS_k, d)$ is the frequency of the SRS SRS_k in the document d. As mentioned earlier, since SRSs are unambiguous due to the presence of related words, the SRS to term translation probabilities would be more specific. Thus, the resulting SRS based smoothed document models will also be more accurate. However, not all portions of a document could be captured by the SRSs alone. First, SRSs which satisfy the two conditions mentioned in section 4.1 only are indexed by the system. Second, the SRSs used may also not be very representative when the document is too short. To handle these problems, we interpolate the SRS translation model with a unigram language model. The accuracy of the SRS translation model is high and the recall of unigram models is good. Thus, interpolating both these models to generate a mixture model seems to be an obvious choice. The famous two stage language model proposed in [Zhai and Lafferty, 2002] is used to smooth the unigram language model and its formula is given below:

$$p(Q/d) = \prod_{q \epsilon Q} \{(1-\gamma)\frac{tf(q,d) + \mu p(q/C)\}}{|d| + \mu} + \gamma p(q/C)\} \tag{8}$$

Where γ and μ are the tuning coefficients and $p(q/C)$ is the background collection model. We call this model the baseline language model following in the lines of Zhou et al.'s work [Zhou *et al.*, 2007a]. The final document model as described earlier is the mixture model of the SRS translation model and the above two-stage language model.

$$p_{b-SRS}(w/d) = (1-\lambda)p_b(w/d) + \lambda p_{SRS}(w/d) \tag{9}$$

Table 1: Top 10 words estimated by the EM algorithm for each SRS

{Space, Program}		{President, of, America}		{Star, War}		{U. S., Technology}	
Word	**Prob.**	**Word**	**Prob.**	**Word**	**Prob.**	**Word**	**Prob.**
space	0.0266	America	0.0312	star	0.0147	technology	0.0231
program	0.0229	president	0.0242	war	0.0123	fighter	0.0173
launch	0.0169	work	0.0129	strategy	0.009	develop	0.0166
technology	0.0161	nation	0.0120	lot	0.0088	Japan	0.0161
orbit	0.0148	United	0.0114	Bush	0.0087	FSX	0.0157
astronaut	0.0148	Bush	0.0108	George	0.0079	U.S.	0.0151
mission	0.0139	love	0.0109	initialize	0.0078	Japanese	0.0146
NASA	0.0136	state	0.0100	permit	0.0070	jet	0.0136
satellite	0.0134	American	0.0097	nuclear	0.0069	industry	0.0135
earth	0.0132	veri	0.0090	office	0.0069	United	0.0133

Where λ is called the SRS translation coefficient and controls the influence of the two components in the mixture model. The mixture model becomes pure SRS translation model if $\lambda = 1$ and it becomes the two-stage language model if $\lambda = 0$.

5 Experiments

5.1 Testing Collection and Queries

Since TREC collections are popular and well studied and many published results exist, we decided to use AP89 collection in our experiments. Early TREC topics are described in multiple sections in terms of *title*, *description*, *narrative* and *concept*. Queries which contain no relevant documents are removed. Following [Berger and Lafferty, 1999] and [Zhou *et al.*, 2007a], we use only the *title* part of the TREC queries, since in real applications queries are similar to titles. The queries are tokenized and the extracted terms are stemmed using the Porter stemmer. Stop words are removed too. Table 2 lists the important statistics of this collection.

Table 2: Statistics of AP89 Collection and Topics 1-50

AP89 Collection	Value
Number of Documents	84,678
Number of unique Words	141,047
Average number of unique words per doc	180.1
Number of unique SRSs in the SRS Index	148,070
Average number of unique SRSs per doc	52.8
Average Query Length	3.4

5.2 Evaluation Metrics

We have followed the TREC convention of using Mean Average Precision (MAP) as our major performance metric. Also, we use the recall at 1000 documents, P@10 and P@100 as our other performance metrics. The formula for non-interpolated average precision as in [Zhou *et al.*, 2007a] is:

$$\frac{1}{|Rel|}\sum_{d \epsilon Rel}\frac{|\{d' \epsilon Rel, r(d') \leq r(d)\}|}{r(d)} \tag{10}$$

where $r(d)$ is the rank of the document d and Rel is the set of relevant documents for a query q. To obtain the MAP score for the collection, we average the non-interpolated average precision across all the queries of the collection.

Table 3: Comparison of the SRS Based Mixture Model with the baseline Two Stage Language Model and the Okapi Model. The collection used is the TREC AP89 collection with topics 1-50.

Metric	Okapi	TSLM	SRS Model	vs. Okapi	vs. TSLM
MAP	0.186	0.187	0.205	+10.22%	+9.63%
Recall	1627	1623	1836	+12.85%	+13.12%
P@10	0.259	0.259	0.262	+1.16%	+1.16%
P@100	0.139	0.139	0.150	+7.91%	+7.91%

5.3 Comparison with the baseline model

The two stage language model (TSLM) mentioned in (8) is the baseline model in our experiments. In addition to TSLM, we also compare our results with the Okapi Model.

The Okapi Model [Robertson *et al.*, 1992] is a popular model and its formula is:

$$sim(Q,d) = \sum_{q \epsilon Q}\left\{\frac{tf(q,d)\log(\frac{N-df(q)+0.5}{df(q)+0.5})}{0.5+1.5\frac{|d|}{avg\_dl}+tf(q,d)}\right\} \tag{11}$$

where,
$tf(q,d)$ is the term frequency of q in document d
$df(q)$ is the document frequency of q
avg_dl is the average document length in the collection

Table 3 shows that the results obtained after performing SRS based context sensitive semantic smoothing on the document models are significantly higher than both the baseline Two Stage Language Model (TSLM) and the Okapi model. In all experiments, the values of γ and μ in the two-stage language model and the value of SRS translation coefficient λ in the SRS model, were set to 0.5, 750 and 0.325 respectively decided empirically. The next section presents the comparison of our SRS model with the MultiWord Expression topic signature model of [Zhou *et al.*, 2007a] which is known to produce more accurate results than the word to word translation model of [Berger and Lafferty, 1999].

5.4 MultiWord Expression (MWE) Context Sensitive vs. SRS Context Sensitive Smoothing

TREC AP89 collection, like any general news collection, has many ambiguous terms. To remove this ambiguity or to include context, both multiword expression and SRSs are used in translation models. However, non-consecutive related words are also present in SRSs unlike multiword expressions.

If we compare the performance of both the models at translation coefficient λ = 1 (i.e. comparison of the SRS translation component of the SRS mixture model with the MWE translation component of the MWE topic signature model), we see that the SRS mixture model shows significant improvements over the MWE topic signature model (see Table 4). This high MAP score of SRS mixture model at λ = 1 indicates that SRS translation component is able to capture more parts of a document than the MWE translation component.

Table 4: Comparison of the SRS Mixture Model with the MWE Topic Signature Model at translation coefficient λ = 1 for both the models. The collection used is the TREC AP89 collection with topics 1-50.

Metric	**MWE Model**	**SRS Model**	**Improv.**
MAP	0.077	0.098	+27.27%
Recall	1289	1413	+9.62%
P@10	0.130	0.168	+29.23%
P@100	0.091	0.104	+14.29%

However, the best performances of both the models i.e. SRS mixture model's performance at λ = 0.325 and MWE topic signature model's performance at λ = 0.3 (see Table 5) are very much comparable indicating the effectiveness of the role played by the baseline TSLM model too in the mixture models.

Table 5: Comparison of the SRS Mixture Model with the MWE Topic Signature Model. For SRS Mixture Model the best MAP value is obtained at λ = 0.325. For MWE Topic Signature Model the best value is obtained at λ = 0.3. The collection used is the TREC AP89 collection with topics 1-50.

Metric	**MWE Model**	**SRS Model**	**Improv.**
MAP	0.204	0.205	+0.49%
Recall	1809	1836	+1.49%
P@10	0.272	0.262	-3.68%
P@100	0.142	0.150	+5.63%

The variance of the MAP with the translation coefficients of both the models is depicted in Figure 2. Since only useful SRSs which satisfy the two conditions described in Section 4.1 are indexed by our system, many parts of the documents are not captured by the translation component of the model. But the baseline two stage language model is able to capture them and thus, possibly, when the weight of the translation

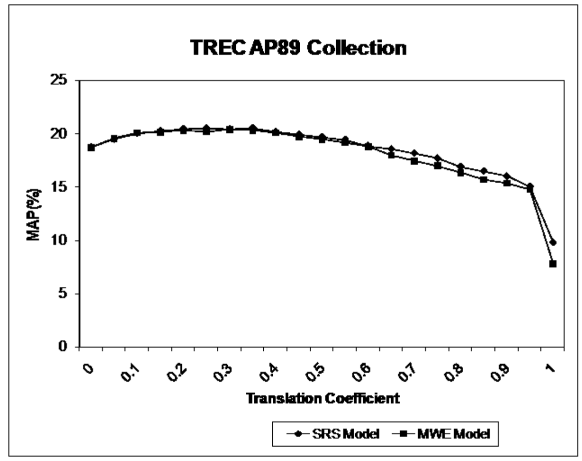

Figure 2: Comparison of the SRS Model with the MWE Topic Signature Model at different values of translation coefficient λ

component in the mixture model is high, the performance goes down. A similar argument for the MWE model could also be proposed.

The results in Table 6 present performance when either the new SRS model or the comparison model (MWE) is picked on each query by looking at which performs better. Although these are oracle results (correctness is known), they interestingly indicate the possibility of combining SRS and MWE models for future developments.

Table 6: Results obtained by the system which picks the best model from the SRS Mixture and the MWE Topic Signature models for each topic. The collection used is the TREC AP89 collection with topics 1-50.

Metric	**MWE**	**SRS**	**SRS+ MWE**	**vs. MWE**	**vs. SRS**
MAP	0.204	0.205	0.217	+6.37%	+5.85%
Recall	1809	1836	1865	+3.1%	+1.58%
P@10	0.272	0.262	0.277	+1.84%	+5.73%
P@100	0.142	0.150	0.153	+7.75%	+2.00%

The current SRS generator module takes a week roughly to convert raw documents of the AP89 collection (84,678 documents) to SRS documents on a desktop computer with 4GB RAM. This module being an offline module doesn't affect the run time performance. Table 7 presents the various statistics.

6 Discussion

As is apparent from Tables 4 and 5 and Figure 2, the SRS based translation model shows high promise in comparison to other topic signature based translation models, notably those based on multiword expressions relying on consecutivity. The MAP score is significantly better (Table 4). As λ tends to 1, the effect of baseline model reduces and the performance of the SRS based translation model shows up. Starting

Table 7: Time and space utilization statistics of SRS System

Name	Value
Processor	Intel Pentium 4 2.4 GHz
RAM	4GB
AP89 collection size	84,678 documents
Time to generate SRS documents from text documents	Around a week
Time to generate word index	10 minutes
Time to generate SRS index	3 hours

from a λ value of about 0.2, the MAP value of the SRS based system continues to be more than that of the multiword based system (Figure 2). One clearly sees the decidedly better performance of the SRS based model when the baseline model is completely absent in the mixture model (at translation coefficient λ value 1).

However, a simple combination of the SRS and the MWE based models gives the best performance indicating that SRSs and MWEs can work in conjunction too (Table 6). Also, the sanity check of comparison with the baseline (two stage language model), of course, shows that introduction of the SRS makes a lot of sense. The scoring of the system over the baseline is very significant (Table 3).

7 Conclusion

We have described here our work on a novel approach to context sensitive semantic smoothing. The approach makes use of semantically related and not-necessarily-consecutive word tuples for document smoothing. These tuples are called SRSs which have been used in machine translation. The SRS based approach consistently outperforms the MultiWord Expression (MWE) based approach on MAP score. However, a simple system which combines the results of the SRS and MWE approaches shows even higher retrieval performance. Our work, thus, shows that the use of NLP inspired patterns in document modeling holds the promise of better IR performance.

Our future work consists of investigating the use of more complex combinations of SRS and MWE based context-sensitive smoothing approaches and using other evaluation metrics too like Normalized Discounted Cummulative Gain (accounts for highly relevant documents appearing lower in the result list). We also intend to reduce the SRS generation time by performing optimizations on the SRS Generator module.

References

[Berger and Lafferty, 1999] Adam Berger and John D. Lafferty. Information retrieval as statistical translation. In *Research and Development in Information Retrieval*, pages 222–229, 1999.

[Charniak, 2000] Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 132–139, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[Gao *et al.*, 2005] Jianfeng Gao, Haoliang Qi, Xinsong Xia, and Jian-Yun Nie. Linear discriminant model for information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 290–297, New York, NY, USA, 2005. ACM.

[Hofmann, 1999] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August 1999.

[Hornby, 2001] A. S. Hornby. *Oxford advanced learner's dictionary of current English*. 2001.

[Khaitan *et al.*, 2007] Sanjeet Khaitan, Kamaljeet Verma, Rajat Mohanty, and Pushpak Bhattacharyya. Exploiting semantic proximity for information retrieval. In *IJCAI '07: Workshop on Cross Lingual Information Access*, 2007.

[Lafferty and Zhai, 2001] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, New York, NY, USA, 2001. ACM.

[Miller, 1994] George A. Miller. Wordnet: a lexical database for english. In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 468–468, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[Mohanty *et al.*, 2005] Rajat Mohanty, Anupama Dutta, and Pushpak Bhattacharyya. Semantically relatable sets: Building blocks for representing semantics. In *MT Summit '05: 10th Machine Translation Summit*, 2005.

[Ponte and Croft, 1998] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM.

[Porter, 1997] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[Robertson *et al.*, 1992] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30, 1992.

[Smadja, 1993] Frank Smadja. Retrieving collocations from text: Xtract. *Comput. Linguist.*, 19(1):143–177, 1993.

[Van Rijsbergen, 1979] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[Zhai and Lafferty, 2001] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, New York, NY, USA, 2001. ACM.

[Zhai and Lafferty, 2002] ChengXiang Zhai and John Lafferty. Two-stage language models for information retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, New York, NY, USA, 2002. ACM.

[Zhou *et al.*, 2006] Xiaohua Zhou, Xiaohua Hu, Xiaodan Zhang, Xia Lin, and Il-Yeol Song. Context-sensitive semantic smoothing for the language modeling approach to genomic ir. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, New York, NY, USA, 2006. ACM.

[Zhou *et al.*, 2007a] Xiaohua Zhou, Xiaohua Hu, and Xiaodan Zhang. Topic signature language models for ad hoc retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19(9):1276–1287, 2007.

[Zhou *et al.*, 2007b] Xiaohua Zhou, Xiaodan Zhang, and Xiaohua Hu. *The Dragon Toolkit Developer Guide*. Data Mining and Bioinformatics Laboratory, Drexel University, 2007.