Language Model: Extension to the Similarity of Non-matching Terms in Retrieval Time

Kian-Lam TAN, Jean-Pierre CHEVALLET, Philippe MULHEM

UJF-Grenoble 1 / UPMF-Grenoble 2 / Grenoble INP / CNRS,
LIG UMR 5217, Grenoble, F-38041, France
{Kian-Lam.Tan,Jean-Pierre.Chevallet,
Philippe.Mulhem}@imag.fr

Résumé : With the explosive growth of online information such as email messages, news articles, scientific literature and many kinds of information on the web, a powerful Information Retrieval System (IRS) is required to manage and provide the best result for users that are both effective and efficient. Most of the IRS use the term intersection approach to develop the matching function. However, this approach does not have a good coverage to solve the problem of term mismatch where different terms are used to describe the same meaning of an object. This paper presents an approach to solve the problem of term mismatch by incorporating the Term Similarity Matrix into the Language Model. Our approach is tested on the Cultural Heritage in CLEF (CHiC) collection which consists of short queries and documents. The results show that the proposed approach is effective and has improved the accuracy in retrieval time.

Mots-clés : Information Retrieval, Term Similarity, Term Mismatch.

1 Introduction

Numerous cultural heritage materials are now accessible through online digital library portals. This made the available resources comparatively burden less to be obtained internationally. However, these materials heavily dependent on the annotator from different cultural heritage institutions and indirectly cause the problem of inconsistency and incompleteness in the metadata. For example if an annotator uses “18th century” to describe an object created in year “1756”, then this annotation is incomplete. If another annotator uses the specific year like “1756” to describe the same object, then two the annotations are inconsistent regarding the content of the annotation. Inconsistency may also refer to the structure of the annotations themselves : for instance some annotators might insert all the descriptions into the same metadata field, and others may split it into multiple metadata fields. In such cases, the information of an object may differ depending on the human annotators.

In this paper, we try to take into account such problems in the context of Information Retrieval (IR). Language Models for IR has been proven that it is a very effective on text retrieval based on [10] and [18]. The extension that we propose in this paper is to integrate the term links (Term Similarity Matrix) into the Language Model based on Dirichlet Smoothing (the most effective Smoothing technique according to [18]). Our proposal has the following advantages: a) it is easy and simple to generate the term links (Term Similarity Matrix) based on statistical information if compare to synthetic queries in [2] or mutual information [7] which we considered as heavy method, and b) it is easy to integrate the term links (Term Similarity Matrix into the Language Model. Moreover, we show that our approach is better than the Dirichlet Smoothing as shown in Section 5.

The rest of this paper is organized as follows. Firstly, we present the state of the art in Section 2. Then, we discuss our idea and approach in Section 3 follow by experiment in Section 4. Finally, we conclude our results and future works in Section 5.

2 State of the Art

In the past, many IR models such as Vector Space Model (VSM) [13, 14], Probabilistic Model [12, 1] and Language Model [10, 18] have been proposed which based on term intersection approach. Term intersection is the approach where both document and query should share the same terms. Although this approach provides a good result in terms of speed and accuracy, but it does not cover the problem of term mismatch which the document does not compromise the same term with the query.

There are a number of approaches that have been proposed to solve the mismatch of terms' problem by using Language Model such as the recent work from [7] and [2] that proposed to use Statistical Translation Model. The main difference between these two works is Berger and Lafferty in using synthetic queries [2] while Karimzadehgan and Zhai proposed to use mutual information to generate term links [1] [7].

[1] Term links refer to the relationship between the term t and t'

2.1 Term Links

As mentioned earlier, our goal of this research is to integrate a Term Similarity Matrix into the Language Model. Before we build the Term Similarity Matrix, we need to find the links between all the terms in the collection naming V this vocabulary.

$$\forall(t, t') \in V, 0 \leq Sim(t, t') \leq 1 \tag{1}$$

1. $Sim(t, t') = 0$, there is no link between the term t and t'
2. $Sim(t, t') < 1$, there is a link between the term t and t'
3. $Sim(t, t') = 1$, there is exact match between the term t and t'

Basically, we make the assumption that two terms are considered link to each other if both terms co-occur in the same context. We use this assumption to build the Term Similarity Matrix. For example, a user is searching for the information about a “temple in ceylon”. The user then submits the query :

$$q = (temple, ceylon)$$

and an IRS considering the documents below:

$d_1 = (temple, india, buddhist, god)$
$d_2 = (india, temple, sri, lanka)$
$d_3 = (roman, temple)$
$d_4 = (india, gautama)$

First and foremost, the IRS assigns a very similar Retrieval Status Value (RSV) to d_1 and d_2, and d_3 (depends on the indexing weights) because these documents contain similar terms as the query which is the “temple”. However, we know that d_3 is surely not relevant since d_3 contains the information of roman temple and not the information of ceylon temple. In addition, we can argue that d_4 is more relevant than d_3 if we compare d_3 with d_4.

If we have the Term Similarity Matrix which contains the link between the term of ”ceylon” and ”india”, then the IRS will return d_1, d_2, d_3 an d_4. Based on this example, it motivates us to consider the non-matching terms by exploiting the term similarity between the query and the document.

Several techniques in [8] have been proposed and the most important methods are : 1) dimension reduction (stemming approach, manual thesaurus, latent semantic indexing), 2) query expansion (automatic, manual or interactive query expansion) and 3) relevance feedback (explicit, implicit or blind feedback). The main differences between these techniques and our proposed approach is that we do not add any extra terms to the query or the document. Our approach only interferes the Relevance Status Value (RSV) value during the matching process.

2.2 Exploiting Term Similarity

2.2.1 Vector Space Models

Crestani [5] proposed a general framework to exploit the term similarity into the matching process based on the variant where $w_d(t)$ is the weight which assigned to term t in the context of document d, and $w_q(t)$ is the weight assigned to term t in the context of query q as shown below :

$$RSV(d,q) = \sum_{t \in q} w_d(t) w_q(t) s \tag{2}$$

In order to visit the non-matching terms in the document, Crestani [5] proposed to exploit the term similarity by utilizing a Sim function. If $Sim(t_i, t_j) = 1$, it means that t_i and t_j are using the same term or we can rewrite it as $t_i = t_j$. If $Sim(t_i, t_j)$ is close to 1, t_i and t_j are strongly related that t_i and t_j can be used to express the same concepts and otherwise is 0.

The proposed idea by Crestani [5] is an extension of the RSV formula (2). The main idea was to extend the matching process that includes a new intermediate term t^* which contains the link between the term t from the document with the term t from the query. Essentially, the term t^* does not need to appear in the query, but as long as the term t^* contains the link with the term t, then it can be used to extend the matching process.

Given a term t from the query [2] means we need to consider all the terms in the document. The main idea of this approach is to consider the term t^* which is the maximum or highest value based on Sim function for a given t in the query as shown below:

$$t^* = \underset{t' \in d}{argmax}(Sim(t, t')) \tag{3}$$

This means that t^* is chosen among the terms that belong to document d because t^* is the maximum or highest value from the Sim function. If

[2] "t from the query" or "from document" refer to the weight of t that is not null in the query, or in the document i.e. $w_q(t) > 0$ or $w_d(t) > 0$

the term t appears in the document d, then the best term t^* should be t itself or we called it as exact match. If t does not appear in the document d, then the weight of t is substituted by a term t^* in the document if there is a similarity value between t and t'. The similarity value should be lower than the value of the exact match which is $t^* = 1$ and it changes the formula in the following way:

$$RSV_{max(q \triangleright d)}(d, q) = Sum_{t \in q} Sim(t, t') w_d(t') w_q(t) \tag{4}$$

or based on (3), it changes the formula in the following way:

$$RSV_{max(q \triangleright d)}(d, q) = Sum_{t \in q}(t^*) w_d(t') w_q(t) \tag{5}$$

In other words, if t^* exists in document d, then the similarity score should be 1 and the formula will be the same as (2). For this reason, we can conclude that the proposed idea by Crestani [5] is an extension of the inner product of the vector d where the term does not appear in the document, but through the Sim function.

Furthermore, Crestani [5] proposed another solution which considers the total value of all non-matching terms in the evaluation of the RSV and this new value $RSV_{tot(q \triangleright d)}$ is defined by:

$$RSV_{tot(q \triangleright d)}(d, q) = Sum_{t \in q} \left(Sum_{t' \in d} Sim(t, t') w_d(t') \right) w_q(t) \tag{6}$$

The main difference for this approach is to use the total value instead of the maximum or highest value for the term t^*. All possible similar terms are used to compute the new weight $w_d^{tot}(t)$:

$$w_d^{tot}(t) = Sum_{t' \in d} Sim(t, t') w_d(t') \tag{7}$$

From the matrix point of view, the computation above equivalent to a matrix product between the similarity matrix Sim and the document vector d and it produces a new extended document vector d_1:

$$d_1 = Sim \times d \tag{8}$$

From the graph point of view, if we consider the Sim matrix as a weighted graph, then this is equivalent to move one step into the graph which is the total value for the term. Hence, it is equivalent to *extent the document* d by using the similarity graph (matrix) and the formula can then be rewritten in:

$$RSV_{tot(q \triangleright d)}(d, q) = (Sim \times d)^\top \times q = d_1^\top \times q \tag{9}$$

2.2.2 Language Models

Recently, Language Models have received considerable attentions because it is based on statistical foundation and a good empirical performance [10][18] and this motivated us to integrate the Term Similarity Matrix into such model. A reference paper from Karimzadehgan and Zhai which proposed to integrate the term similarity (translated into probabilities) into the Language Model based on Statistical Translation Model [7]. In addition, they rely on data from the corpus itself like synthetic queries as in [2] and not from other resources. In some ways, we can consider that their proposal is related to the second proposition from Crestani [5] which the idea is to consider the similarity between each term from query term and the terms from document. The results obtained by Karimzadehgan and Zhai [7] showed that the integration between the term similarity and Language Model is more efficient and more effective than the existing approaches in Information Retrieval.

However, Karimzadehgan and Zhai [7] noticed that the self-translation probabilities lead to non-optimal retrieval performance because it is possible that the value of $P(w|u)$ is higher than $P(w|w)$ for a term w. In order to overcome this problem, Karimzadehgan and Zhai [7] defined a parameter to control the effect of the self-translation.

In a nutshell, we can remark that 1) the normalization of the mutual information is rather artificial and required a parameter to control the effect of the self-translation, and 2) the regularization of the initial transition probabilities may look uncertain.

3 Proposal

As mentioned earlier, our goal is to integrate the Term Similarity Matrix into the Language Model. After the reviews of Crestani [5], Karimzadehgan and Zhai [7] and Zhai [17], we had considered the problems and propose to use the approach as shown below:

- We propose to use the maximum or highest value instead the total value from the term similarity between the terms from query with the terms from document. Besides, we only consider the point we view of a query if we cannot find a term in the document, then we consider the closest semantic terms from the document.
- We propose to use statistical approach rather than probability approach in order to avoid the value of $P(w|u)$ is higher than $P(w|w)$ for a term w obtained by Karimzadehgan and Zhai [7].

3.1 Extended Dirichlet Smoothing

The Language Model approach in IR was proposed by Ponte and Croft [10]. The basic idea of Language Model is to assume that a query q which is generated by a probabilistic model based on a document d as shown below:

$$P(d|q) \propto P(q|d).P(d) \tag{10}$$

$P(q|d)$ is the query likelihood for the given document d matches with the query q. If we consider that every document is equally relevant to any other query, then we can discard the $P(d)$ parameter and we can rewrite the above formula as shown below:

$$P(d|q) = \sum_{w_i \in C} c(w, d).P(w|d) \tag{11}$$

where $c(w, q)$ is the count of words w in query q and C is a set of vocabulary. Based on the multinomial distribution, the simplest way to estimate $P(w|d)$ is through the maximum likelihood estimator:

$$P_{ml}(w|d) = \frac{c(w, d)}{|d|} \tag{12}$$

where $|d|$ is the total length of the document d. Due to the data spareness problem, the maximum likelihood estimator directly assign $null$ to the unseen words in a document. Smoothing is a technique to assign extra probability mass to the unseen words in order to solve this problem.

Basically, Dirichlet [18] is one of the smoothing technique which based on the principle of adding an extra pseudo term frequency which is $\mu P(w|C)$. The Dirichlet smoothing is obtained by taking into account the extra pseudo term frequency distribution:

$$P_{\mu}(w|d) = \frac{c(w; d) + \mu P(w|C)}{\sum_w c(w; d) + \mu} \tag{13}$$

The main idea of this research is to integrate the Term Similarity Matrix into the current Dirichlet formula. Firstly, we need to assume that a term w is $w' \in d$ can play the role of w where w is $w \in q$ during the matching process. More specifically, we consider that if w does not occur in the initial document d, but it occurs in the *document* d_{ext}, which is the result of the extension of d according to the query and some knowledge[3]. Then,

[3] The knowledge refers to the Term Similarity Matrix

the probability of the term will define according to the extended document d_{ext}.

The knowledge assumes to form a symmetrical similarity function which is $Sim : V \times V \rightarrow [0,1]$, that denotes the strength of the similarity between two terms from the vocabulary (the larger the value, the higher the strength). We propose that : $\forall w \in V, Sim(w, w') = 1$ if exact matching between w with w', and $\forall w \in V, Sim(w, w') = 0$ if w does not contain any link with w'.

In order to avoid any complex extensions (see the state of the art), we define the following constraints :

– one query term w must only impact occurrences of one document term w' ;

To achieve this, we use some simple and sensible heuristics:

1. If a query term w occurs in a document d, then the term will not change the length of the document.
2. If a query term w does not occur in a document d but the term w contains a link with w' (term from document), then we define $w'' = argmax_{w' \in d, w' \neq w} Sim(w, w')$ as the term from the document will serve as the basis count of the pseudo occurrences of w in d as $c(w''; d).Sim(w'', w)$. This pseudo occurrences of the term w'' are then included into the size of the extended document.
3. If a query term w does not occur in the document and does not contains any link, then it's occurrences is counted in the extended document.

Eventually, using usual set of notations for the terms that occur in the document and the query, then the new length of the document ($|d_{ext}|$) is:

$$|d_{ext}| = \sum_{w \in d \cap q} c(w; d) + \sum_{w'' \in d \backslash q; Sim(w, w'') \neq 0} c(w''; d).Sim(w'', w) \\ + \sum_{w' \in d \backslash q; Sim(w, w') = 0} c(w'; d)$$

with w” defined above for one query term w so that:

$$w'' = argmax_{w' \in d, w' \neq w} Sim(w, w') \tag{14}$$

Using the fact above, the expression of $|d_{ext}|$ can be easily simplified into:

$$|d_{ext}| = |d| + \sum_{w'' \in d \backslash q; Sim(w, w'') \neq 0} c(w''; d).Sim(w'', w) \tag{15}$$

Remind that our proposal is to extend the document according to the query. With all the elements described above, the extended Dirichlet Smoothing leads to the following probability for the term w of the vocabulary V in the document extended d_{ext} according to a query q, noted that $p_{\mu}(w|d_{ext})$ is defined as:

1. if $w \in d \cap q$:

$$P_{\mu}(w|d_{ext}) = \frac{c(w;d) + \mu P(w'|C)}{|d_{ext}| + \mu} \tag{16}$$

2. if $\exists w'' \in d \setminus q; Sim(w, w'') \neq 0$:

$$P_{\mu}(w|d_{ext}) = \frac{c(w";d).Sim(w,w") + \mu P(w"|C)}{|d_{ext}| + \mu} \tag{17}$$

with $w'' = argmax_{w' \in d, w' \neq w} Sim(w, w')$.

3. if $\nexists w'' \in d \setminus q; Sim(w, w'') \neq 0$

$$P_{\mu}(w|d_{ext}) = \frac{c(w;d) + \mu P(w|C)}{|d_{ext}| + \mu} \tag{18}$$

with $w'' = argmax_{w' \in d, w' \neq w} Sim(w, w')$.

In the specific case when all the query terms from q occur in the document d, the first case in the above is used where $|d_{ext}| = |d|$ leads to $p_{\mu}(w|d) = p_{\mu}(w|d_{ext})$.

3.2 Term Similarity Matrix Based on Statistical Approaches

In this section, we propose an easier and a more efficient way to compute the Term Similarity Matrix based on statistical approach which can have a better coverage.

Similarity between terms can be represented in a variety ways. In our approach, we used Confidence Coefficient, Tanimoto Similarity, Dice Coefficient, Cosine Similarity and Overlap Coefficient to generate the statistical information [11][6]. The Confidence Coefficient between term w_i and w_j are calculated as follows:

$$Sim_{conf}(w_i, w_j) = \frac{n(w_i \cap w_j)}{n(w_i)} or \frac{n(w_i \cap w_j)}{n(w_j)} \tag{19}$$

where $n(w_i)$ is the number of term(w_i) in the corpus, and $n(w_i \cap w_j)$ is the number of terms that term w_i co-occur together with w_j in the corpus.

The Tanimoto Similarity between term w_i and w_j are calculated as follows:

$$Sim_{tani}(w_i, w_j) = \frac{n(w_i \cap w_j)}{n(w_i) + n(w_j) - n(w_i \cap w_j)} \tag{20}$$

where $n(w_i)$ is the number of term(w_i) in the corpus, and $n(w_i \cap w_j)$ is the number of terms that term w_i co-occur together with w_j in the corpus.

The Dice Coefficient [6] between term w_i and w_j are calculated as follows:

$$Sim_{dice}(w_i, w_j) = \frac{2n|w_i \cap w_j|}{n(w_i) + n(w_j)} \tag{21}$$

where $n(w_i)$ is the number of term(w_i) in the corpus, and $n(w_i \cap w_j)$ is the number of terms that term w_i co-occur together with w_j in the corpus.

The Cosine Similarity between term X and Y is represented using a dot product and magnitude as follows:

$$Sim_{cosine}(w_i, w_j) = \sqrt{\frac{n(w_i \cap w_j)}{n(w_i).n(w_j)}} \tag{22}$$

where $n(w_i)$ is the number of term(w_i) in the corpus, and $n(w_i \cap w_j)$ is the number of terms that term w_i co-occur together with w_j in the corpus.

The Overlap Coefficient between term X and Y are calculated as follows:

$$Sim_{over}(w_i, w_j) = \frac{n(w_i \cap w_j)}{min(n(w_i), n(w_j))} \tag{23}$$

where $n(w_i)$ is the number of term(w_i) in the corpus, and $n(w_i \cap w_j)$ is the number of terms that term w_i co-occur together with w_j in the corpus.

4 Experiments

4.1 Data Set

We use the CHiC 2012 to test our proposed idea. CHiC 2012 contains fifty queries and one million documents. The proposed model is a generic solution to all application domains. However, CHiC 2012 was chosen as our test collection because the proposed model is more dedicated to the subject of heritage. By using CHiC, the proposed model returns the best results and thus it could be the baseline benchmark when thus generic model is applied to another application domains.

In this corpus, the metadata inside the documents is quite variable from large to limited data. We use external resources such as Wikipedia to generate the Term Similarity Matrix. Our idea is to compute all the terms which co-occur in the Wikipedia. We use the English Wikipedia (version 2012-01-01) which contains 3.835 million articles in the corpus. For this paper, we only use the first paragraph of each article from the Wikipedia to generate the Term Similarity Matrix. Basically, the first paragraph of each article in the Wikipedia pertains the most critical idea of an article and it can stand on it owns as a concise version of this article according to the guideline from Wikipedia. In the experiments, we only use the title without any description from the queries. As for pre-processing, we remove all the stop words which contains 571 words and non-character, and apply the Porter Stemming. Besides, we convert all the upper case to lower case in order to reduce the term dimension.

All the experiments are done by using the XIOTA engine [4]. The performance is measured by Mean Average Precision (MAP). The optimal value for Dirichlet prior smoothing for baseline is 100 and 350 for all the Extended Dirichlet. Besides, we applied student's paired t-test (at the $p < 0.06$) to assess the significance of the difference measurement between the several types of statistic approach.

Table 1, shown clearly that our approach outperforms the baseline result. The most statistical significant improvement is with the Extended Dirichlet and Dice Coefficient from 0.5273 to 0.5450 at Table 1 while the most depreciation is with the Extended Dirichlet with Overlap Coefficient. The reason for these bad result for (Extended Dirichlet with Overlap Coefficient) is that most of the non-null values of the similarity matrix equal "1" which is abnormal because the value of "1" should represent exact match. Overall, 16 queries show increments, 8 queries show fluctuations and 11 queries remain the same as shown in Figure 1. The most significant increment is in Query 25 which increase around $+2008\%$ from 0.0025 to 0.0547 in terms of Average Precision(AP). The most significant decrements is in Query 28 which decrease around -13.33% from 0.0015(AP) to 0.0013(AP). We may notice that these extreme variation occur at rather low values of AP.

TABLE 1 – Performance with Various Types of Statistic from the First Paragraph of the Articles from Wikipedia (* = statistical significance at $p < 0.06$ using the Student's Paired T-Test)

Types of Approaches	MAP	MAP Gain/Lost
LM + Dirichlet (BL)	0.5273	
LM + Extended Dirichlet + Confidence Coefficient	0.5196	-1.48%
LM + Extended Dirichlet + Tanimoto Similarity	0.5395	+2.31%
LM + Extended Dirichlet + Dice Coefficient	**0.5451***	**+3.38%**
LM + Extended Dirichlet + Cosine Similarity	0.5418	+2.75%
LM + Extended Dirichlet + Overlap Coefficient	0.4929	-6.97%

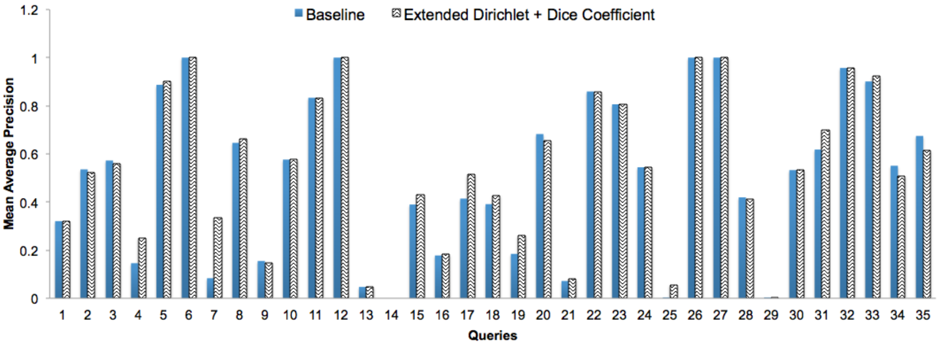

FIGURE 1 – Comparison between the baseline and Dice Coefficient result.

5 Conclusion and Future Work

We have presented a model to exploit the term similarity of non-matching terms during the retrieval time. Our experiments result indicate that the propose approach which is Term Similarity Matrix based on the statistical approach is more efficient and effective than the term intersection approach. For future work, we would like to compute more Term Similarity Matrix from other external resources and not only limited to Wikipedia. If we have more Term Similarity Matrix from different resources means we have higher degree of knowledge to build the link between two different terms.

Références

[1] Gianni Amati and Cornelis Joost van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transaction on Information Systems*, 20(4) :357–389, October 2002.

[2] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 222–229, New York, NY, USA, 1999. ACM.

[3] Guihong Cao, Jian-Yun Nie, and Jing Bai. Integrating word relationships into language models. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 298–305, New York, NY, USA, 2005. ACM.

[4] Jean-Pierre Chevallet. X-iota : An open xml framework for ir experimentation. In SungHyon Myaeng, Ming Zhou, Kam-Fai Wong, and Hong-Jiang Zhang, editors, *Information Retrieval Technology*, volume 3411 of *Lecture Notes in Computer Science*, pages 263–280. Springer Berlin Heidelberg, 2005.

[5] Fabio Crestani. Exploiting the similarity of non-matching terms at retrieval time. *Journal of Information Retrieval*, 2 :25–45, 2000.

[6] William B. Frakes and Ricardo Baeza-Yates. *Information Retrieval : Data Structures and Algorithms*. Prentice Hall PTR, June 1992.

[7] Maryam Karimzadehgan and ChengXiang Zhai. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 323–330, New York, NY, USA, 2010. ACM.

[8] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[9] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Wordnet : An on-line lexical database. *International Journal of Lexicography*, 3 :235–244, 1990.

[10] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. pages 275–281, 1998.

[11] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[12] S. E. Robertson. Overview of the okapi projects. *Journal of Documentation*, 53(1) :3–7, 1997.

[13] Gerard Salton. The smart project in automatic document retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 356 – 358, Chicago,

Illinois, United States, 1991.

[14] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. In *INFORMATION PROCESSING AND MANAGEMENT*, pages 513–523, 1988.

[15] W. Tannebaum and A. Rauber. Acquiring lexical knowledge from query logs for query expansion in patent searching. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 336–338, 2012.

[16] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.

[17] ChengXiang Zhai. *Statistical Language Models for Information Retrieval*. Now Publishers Inc., Hanover, MA, USA, 2008.

[18] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval.