Facilitating Query Decomposition in Query Language Modeling by Association Rule Mining Using Multiple Sliding Windows

Dawei Song[1], Qiang Huang[1], Stefan Rüger[1], and Peter Bruza[2]

[1] Knowledge Media Institute, The Open Univeristy, Milton Keynes, UK
[2] Queensland University of Technology, Australia
{d.song,q.huang,s.rueger}@open.ac.uk,p.bruza@qut.edu.au

Abstract. This paper presents a novel framework to further advance the recent trend of using query decomposition and high-order term relationships in query language modeling, which takes into account terms implicitly associated with different subsets of query terms. Existing approaches, most remarkably the language model based on the Information Flow method are however unable to capture multiple levels of associations and also suffer from a high computational overhead. In this paper, we propose to compute association rules from pseudo feedback documents that are segmented into variable length chunks via multiple sliding windows of different sizes. Extensive experiments have been conducted on various TREC collections and our approach significantly outperforms a baseline Query Likelihood language model, the Relevance Model and the Information Flow model.

Key words: Association Rule, Term Relationship, Query Expansion, Document Segmentation

1 Introduction

Recent studies in language modeling (LM) have tried to exploit relevance feedback documents to establish an improved query model via a model-based approach [9, 10, 16]. One example is the Relevance Model (RM) [10], which estimates the joint probability of observing a term w in the vocabulary together with query topic $Q = \{q_1, \cdots, q_{|Q|}\}$. The assumption of independence among query terms has been made to reduce the complexity of computation. This, however, neglects the effect of the relationships between terms in determining the query language model.

More recent research [15, 13, 3, 5, 12] tries to incorporate term relationships or dependencies, for example, grammatical links [6], co-occurrence and WordNet relations [5], in LM. It also has been shown in [5] that combining multiple types of term relationships, e.g., co-occurrences and WordNet relations, leads to improvement on average precision over the use of different types individually.

Furthermore, there has been a trend of decomposing a query into different combinations (subsets) of query terms, and using term relationships derived from the subsets of query terms rather than traditional pairwise term co-occurrences. These automatically derived term relationships are in higher order, in the sense that the information is flowing from a set of terms to another term (e.g. "(java, computer) $\rightarrow$ programming"). "java" and "computer" are combined to form a context-dependent premise for the derivation of "programming". Song and Bruza [15] propose an information flow model to capture the relationships, and in [13], Pickens and MacFarlane build a term context model based on a maximum entropy algorithm to estimate the co-occurrence of terms in documents with the query topic. The work in [12] expands the approach used in [13], and decomposes the query topic into "latent" concepts, which consist of the combinations of query terms. In [3], high-order inferential term relationships extracted by the information flow approach [15] have been employed in a LM framework combining the effects of information flows from different subsets of query terms.

Essentially, the Information Flow approach [15] is based on a lexical semantic space model, namely Hyperspace Analogue to Language (HAL). The HAL space is constructed by moving a fixed length sliding window over the corpus by a one term increment. All terms within the window are considered as co-occurring with each other with strengths inversely proportional to the distance between them. After traversing the corpus, numeric vectors representing the concepts (terms) are produced. Arbitrary terms (e.g., "Java" and "computer") that are related to each other (but not necessarily the syntactically valid phrases) can be combined to form a new concept, also represented as a vector, by a weighted addition of the underlying vectors of the terms. The information flow between two concepts is then computed by comparing their underlying vectors.

Despite its good performance, re-loading and manipulating vectors in the pre-computed HAL space, which is normally very large, for each query session may potentially lead to a high computational overhead. In particular, for query decomposition, the expensive information flow computation process (sequential scan of the vocabulary to compare each vector in the HAL space with the vector representing a subset of query terms) has to be performed for $2^{|Q|}$ times, i.e., for each of the subsets of query terms. Indeed, as a consequence, in both [15] and [3] the query decomposition was not actually performed. It was instead approximated by computing information flows only once from the whole set of query terms only. Moreover, the fixed sized sliding window approach used in HAL is less flexible to encode various levels of associations between terms.

This paper aims to further advance the trend of using query decomposition and high-order term relationships in query language modeling, by developing a novel method to overcome the two aforesaid limitations.

Firstly, we propose to use association rule mining which is a popular and well researched method for discovering interesting relations between variables in large databases. Compared with the information flow approach, the association rule mining shows a strong ability to capture the high-order term relationships from different subsets of query terms in one go, thus truly realizing the idea of query decomposition. It also does not need any training data (as in the MRF method) or a pre-computed semantic space.

Secondly, we propose dividing the documents into variable length segments through multiple sliding windows of different sizes to perform association rule mining. Using shorter segments instead of the whole documents will reduce the computational load of association rule mining. On the other hand, using viable length windows for document segmentation enables different levels of term associations generated from different sized segments to be taken into account in a mixture model. The segmentation-based approaches [11, 2, 4] have had a proven track record particularly in passage and sentence retrieval. However, to our knowledge, there has not been an approach to the use of multiple-sized sliding windows for query language modeling.

In this paper, we build a novel framework that integrates the advantages of association rule mining, multiple window segmentation and query decomposition to derive higher-order term relationships for query language modeling. Fig. 1 shows our overall approach, with pseudo feedback documents and query topic as input, and the generated query model as output.

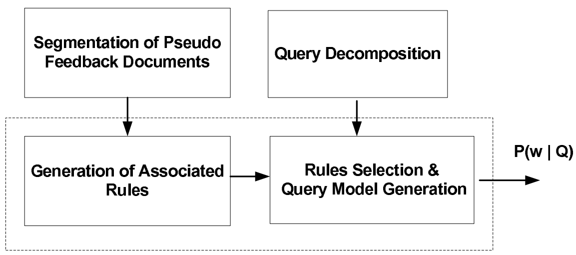


Fig. 1. Structure of Theory

Extensive empirical evaluation has been conducted to compare the effectiveness of our approach with a baseline language model, the Relevance Model and the Information Flow approach. Our approach has demonstrated a better performance than these existing models.

2 Generation of Association Rules from Documents

Mining association rules is an important technique for discovering meaningful patterns in transaction databases. Formally, the problem can be formulated as follows [1]. Let $I = \{i_1, i_2, \cdots, i_n\}$ be a set of n binary attributes called *items*. Let $D = \{t_1, t_2, \cdots, t_m\}$ be a set of transactions called the *database*. Each transaction in D has a unique transaction ID and contains a subset of the items in I [7]. An association rule is a rule of the form $X \Rightarrow Y$, where $X, Y \subseteq I$, and X, Y are two disjoint sets of items. It means that if all the items in X are found in a transaction then it is likely that the items in Y are also contained in the transaction. The sets of items X and Y are respectively called the *antecedent* and *consequent* of the rule [7]. To select interesting rules from the set of all possible rules, constraints on various measures of significance and strength can be used. The best-known constraints are minimum thresholds on support and confidence.

$$supp(X \Rightarrow Y) = supp(X \cup Y) = \frac{C_{XY}}{M} \tag{1}$$

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}, \tag{2}$$

where C_{XY} is the number of transactions which contain all the items in X and Y, and M is the number of transactions in the database.

Support, in Equation 1, is defined as the fraction of transactions in the database which contain all items in a specific rule [14]. *Confidence*, in Equation 2, is an estimate of the conditional probability $P(E_Y|E_X)$, where E_X (E_Y) is the event that X (Y) occurs in a transaction [8].

According to the above definitions, by considering a term as an item and a text fragment, e.g., a sentence, a paragraph, a document, or a fixed sized window, as a transaction, we can easily apply association rule mining to the discovery of high-order term associations (i.e., the association rules between any subset (Q_j) of query terms $\{q_{j_1}, \cdots, q_{j_m}\}$ ($m = |Q_j|$) and the terms from vocabulary.

anti-missil $\Rightarrow$ *direct* (0.076, 36.67)
reform welfar $\Rightarrow$ *competit* (0.312, 177.16)
initi research defens $\Rightarrow$ *contract* (0.162, 1788)
high-combustion fuels create $\Rightarrow$ *laser* (0.03265, 61.11)

Fig. 2. Association Rules

Figure 2 shows some example association rules between various combinations of query terms on the left hand side and a word on the right hand side. The Porter stemmer has been used on these words . The numbers in the bracket following each rule are its *Support* and *Confidence* values respectively.

For association rules, a minimum support threshold is used to select the most frequent item combinations called frequent item sets. The computational complexity of finding these frequent item sets, in the worst case, can be exponential with respect to the number of items. Obviously, using the whole documents as transactions implies high computation cost. Segmenting a long document into shorter chunks can reduce such cost. There has been work carried out in passage and sentence retrieval. However, passages, which are often still quite long (e.g., minimum 50 terms [11]), may contain "noisy" information and lead to computational overhead. On the other hand, the use of sentences may miss some useful relationships between terms. In addition, using passages or sentences directly as transactions may also lead to the data sparseness problem. In this paper, we overcome these problems by segmenting documents into chunks using multiple sliding windows.

3 Document Segmentation

In this paper, we propose the use of multiple sliding windows. Each window slides over the documents with 1/3 overlapping of the length of the window. Eventually, we segment the pseudo feedback documents into chunks of variable lengths.

Documents:	□□□□□□□□□□□□□□□□□□□□□□□□□□ · · · □□□□□□□□□□
Chunk_1:	□□□□□□□□□□
Chunk_2:	□□□□□□□□□□
Chunk_3:	□□□□□□□□□□
· · ·	· · ·
Chunk_M:	□□□□□□□□□□

Fig. 3. Segmentation of Document

Fig. 3 illustrates the process of document segmentation by dividing the whole document into overlapped chunks using a sliding window. The generation of the overlapped chunks obviously increases the number of segments extracted from the pseudo feedback documents, which can reduce the problem of data sparsity to some extent, compared with the use of non-overlapping passages and sentences for retrieval. To alleviate the potential noise carried by longer windows and the potential missing information caused by shorter windows, we apply multiple windows to generating chunks in different lengthes. In our experiments, we tested 7 windows ranging from 15 to 45 terms with a 5 term increment.

4 Rule Selection and Query Model Generation

Association rule mining is then applied in the segmented chunks of the pseudo relevance feedback documents to discover terms associated with different combinations of query terms.

Instead of deriving association rules from a query as a whole, the query is first decomposed into all the possible combinations of query terms. Consequently, an example query $Q = \{q_1, q_2\}$ can be expanded to a list of subsets of query terms, $Q' = \{\{q_1\}, \{q_2\}, \{q_1, q_2\}\}$. A concrete example is shown in figure 4.

$$\begin{array}{ccc} \mathit{Query} & \rightarrow & \mathit{decomposed\ Query} \\ \{\mathit{theory}, \mathit{derivation}\} & \rightarrow & \{\{\mathit{theory}\}, \{\mathit{derivation}\}, \{\mathit{theory}, \mathit{derivation}\}\} \end{array}$$

Fig. 4. Example of the query decomposition

Based on the query decomposition, we refine the association rule mining process by selecting those rules derived from any subset of query terms. The process brings two advantages. One one hand, it collects the rules related to any portion of the query instead of the whole query only. On the other hand, as the association rule mining is incremental, there is no additional computational costs for generating the rules from subsets other than the whole query.

Fig. 5 shows some refined rules from different subsets of query terms. In general, the *Confidence* value can effectively represent how good a captured rule is. As shown in Figure 5, the term combination "*high-combustion fuels hydrogen*" implies "*laser*" in a higher confidence than the others.

high-combustion fuel create ⇒ *laser* (0.03265, 61.11)
high-combustion fuel hydrogen ⇒ *laser* (0.01929, 100.00)
high-combustion fuel ⇒ *laser* (0.01484, 47.62)
high-combustion fuel energy ⇒ *laser* (0.02671, 56.25)
high-combustion create hydrogen ⇒ *laser* (0.01336, 100.00)

Fig. 5. Association Rules

After the associated rules derivable from all the subsets of query terms are obtained, the query model $P(w|Q)$ can be generated as follows.

$$P(w|Q) = \sum_{Q_j \in Q'} P(w|Q_j, Q)P(Q_j|Q) \tag{3}$$

Equation 3 shows a model by mixing the probability of term w given a specific subset of query terms Q_j and Q, weighted by a prior $P(Q_j|Q)$. By assuming $P(w|Q_j, Q) \approx P(w|Q_j)$, we can obtain a simplified version which has been used in [3]:

$$P(w|Q) = \sum_{Q_j \in Q'} P(w|Q_j)P(Q_j|Q) \tag{4}$$

Note that, in [3], the effect of Q_j was not actually implemented, due to the time consuming information flow computations for all the Q_j. Instead, Equation 4 was approximated by the information flows from the whole query Q only.

In addition, we have tested a number of ways for determining the prior distribution $P(Q_j|Q)$, e.g., based on the length of Q_j and the average IDF value of the terms in Q_j. However, our prior experiments show that they are not much of an improvement over the simple uniform distribution. Therefore, in our experiments, $P(Q_j|Q)$ is assumed to be uniform:

$$P(Q_j|Q) = \frac{1}{|Q'|} \tag{5}$$

Equation 4 is then rewritten as:

$$P(w|Q) = \sum_{Q_j \in Q'} P(w|Q_j)/|Q'| \tag{6}$$

Table 1. Test Collections and Query Topics

Coll.	Description	Size (MB)	# Doc.	Vocab.	Query	Q.fields	Q.length (words)
AP89	Associated Press (1989) Disk 1	254	84,678	137,728	1–50	title	3.2
AP88–89	Associated Press (1988–1989) Disk 1,2	492	164,597	254,872	101–150	title	3.6
					151–200	title	4.3
WSJ90–92	Associated Press (1990–1992) Disk 2	242	74,520	121,944	201–250	desc.	8
SJM	San Jose Mercury News (1991) Disk3	287	90,257	146,512	51–100	title & desc.	12.2

In the process of computing the conditional probability $P(w|Q_j)$, we propose using the $Confidence$ values of those associated rules from Q_j.

$$P_{AR}(w|Q_j) = \frac{Conf(Q_j \Rightarrow w)}{\sum_{w'} Conf(Q_j \Rightarrow w')} \tag{7}$$

To derive the "new" smoothed model, a linear mixture can typically be used. In this paper, we also use the method to mix the derived query model from association rule mining with an original query model $P_O(q_i|Q)$, where q_i is a term in the original query.

$$P_O(q_i|Q) = \frac{QTF * IDF(q_i)}{\sum_{j\in 1\cdots|Q|} QTF * IDF(q_j)} \tag{8}$$

where QTF is the number of q_i occurring in the query.
The smoothed query model can then be derived:

$$P_{NEW}(w|Q) = \lambda P(w|Q) + (1-\lambda) P_O(w|Q) \tag{9}$$

5 Empirical Evaluation

5.1 Data

The experiments are conducted using various TREC collections and query topics shown in Table 1. Different fields of the five topic sets are used in different experiments to verify the robustness of our method with respect to different average query lengths. All documents have been pre-processed in a standard manner: terms are stemmed and stop words are removed.

5.2 Experimental Setup

In our experiments, the Lemur Toolkit was used to construct the baseline. For association rule mining, the Apriori algorithm implemented in the WEKA toolkit was adapted with the granularity of transactions set to be at the chunk level. We use the top 35 documents as pseudo feedback documents, and the top 100 terms

Table 2. Comparison between QL, RM, IF and AR

(a) Experimental results on AP89 collection for queries 1–50 (title)

	QL	RM	IF	AR	AvgPr change (% over QL)	AvgPr change (% over RM)	AvgPr change (% over IF)
AvgPr	0.1970	0.2270	0.2664	0.2731	+38.6**	+20.3**	+2.5
Recall	1702	2312	2372	2367			

(b) Experimental results on AP88-89 collection for queries 101–150 (title)

	QL	RM	IF	AR	AvgPr change (% over QL)	AvgPr change (% over RM)	Avgr change (% over IF)
AvgPr	0.2338	0.3069	0.3185	0.3287	+40.6**	+7.1*	+3.2*
Recall	3160	3910	3900	3935			

(c) Experimental results on AP88-89 collection for queries 151–200 (title)

	QL	RM	IF	AR	AvgPr change (% over QL)	AvgPr change (% over RM)	Avgr change (% over IF)
AvgPr	0.3063	0.3471	0.3942	0.4081	+33.2**	+17.6**	+3.5*
Recall	3319	3566	3841	3793			

(d) Experimental results on WSJ90-92 collection for queries 201–250 (description)

	QL	RM	IF	AR	AvgPr change (% over QL)	AvgPr change (% over RM)	Avgr change (% over IF)
AvgPr	0.2366	0.2403	0.2673	0.2846	+20.3**	+18.43**	+6.5*
Recall	978	990	1015	1038			

(e) Experimental results on SJM collection for queries 51–100 (title & description)

	QL	RM	IF	AR	AvgPr change (% over QL)	AvgPr change (% over RM)	Avgr change (% over IF)
AvgPr	0.2105	0.2154	0.2201	0.2372	+12.7*	+10.12*	+7.8*
Recall	1460	1486	1488	1498			

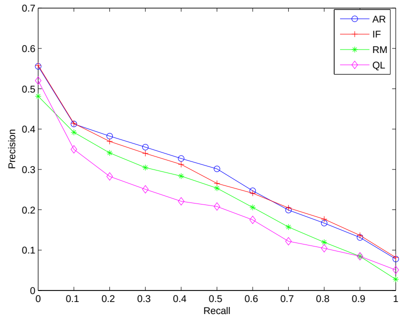
$*$ indicates the difference is statistically significant at the level of $p-$value < 0.05
$**$ indicates the difference is statistically significant at the level of $p-$value < 0.01

from the new query model are selected. Our experiments show little variation in performance when λ is more than 0.9.

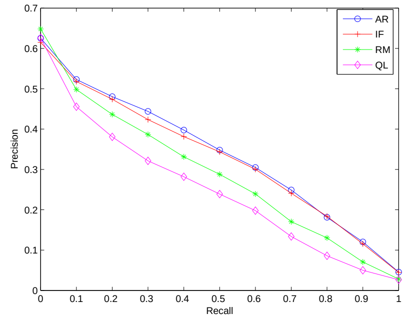
Our method (AR) is compared with a baseline language model, namely the Query Likelihood (QL) model, the Relevance Model (RM), and the language model based on Information Flow (IF). The effectiveness indicators are the standard non-interpolated average precision (AvgP) and recall, which are calculated based on 1000 retrieved documents for each query. We also perform the t-test to measure the statistical significance of performance improvements.

5.3 Result Analysis

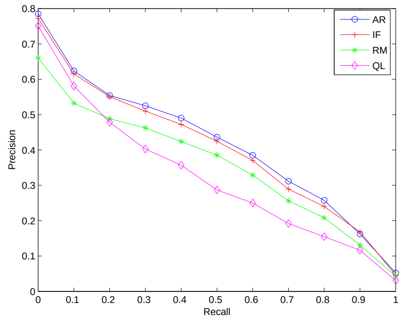
Tables 2(a), 2(b) and 2(c) show the retrieval performance of the four models under comparison on the AP89 and AP8889 collections using the title field of three query sets (average query length: 4). Our approach (AR) shows statistically significant improvements over the Query Likelihood model (QL) by more than 30% (38.6%, 40.6% and 33.2%), and over the Relevance Model by at least 9% (23.4%, 9.3% and 19.9%). Our approach also improves recall over the QL and RM. As shown in Figures 6(a), 6(b) and 6(c), our approach generates better precision than QL and RM at almost all the recall points.



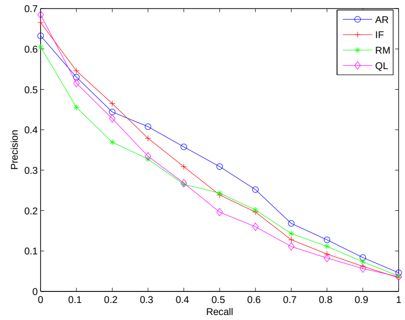
(a) Query 1–50 on AP89



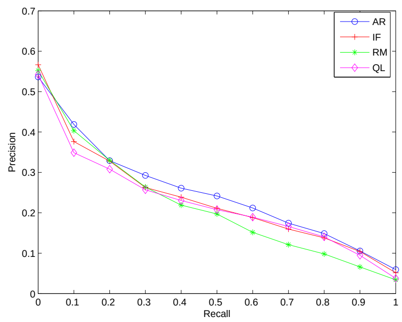
(b) Query 101–150 on AP88–89



(c) Query 150–200 on Collection AP88–89



(d) Query 201–250 on WSJ90–92



(e) Query 51–100 on SJM

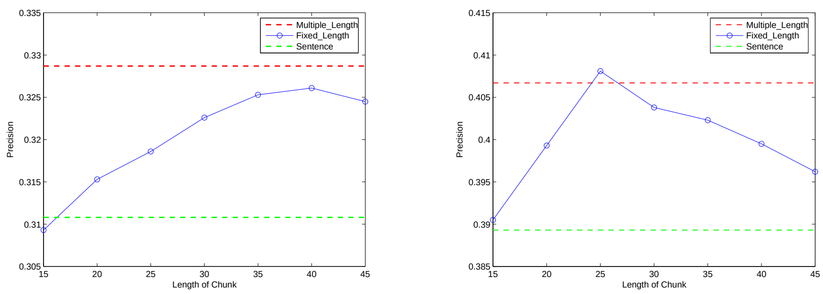
Fig. 6. Precision-recall Curves

Table 2(d) and Fig. 6(d) list the results on the WSJ collection using the description field of topics 201-250 (average query length: 8). Our approach also shows significant improvements in the average precision over the QL and RM, by 20.3% and 17.8% respectively.

Further, the experimental results on the SJM collection using a longer query set (title and description fields of topics 51-100, average query length: 12.2) are shown in Table 2(e) and Fig. 6(e). Again, significant improvements (12.7% and 10.38%) in average precision have been achieved, although not as much as the improvements obtained for the shorter queries. This is due to the query length. In general, the longer the query is, the more useful information it may have contained. Thus, it is reasonable that, for longer queries, a better baseline was also obtained. However, even in this case, our approach still shows its strong ability to capture the relationships between query terms and words in pseudo feedback documents.

Our approach has also shown improvements over the Information Flow based language model in all the experiments. Moreover, more improvements are obtained on longer queries. This reflects the effects of the query decomposition, i.e., the consideration of the contributions from any parts of the query will lead to improvement of retrieval effectiveness.

Fig. 7 compares the retrieval performance with the use of sentences as chunks as well as the use of individual sliding windows. For the former, we split the pseudo feedback documents into sentences based on the punctuation, such as full stops. The performance of the use of sentences is only slightly better than the 15-sized window, but lower than the others. The use of multiple-length chunks proves to be more effective than the use of individual fixed-sized windows.



(a) Query 101–150 on AP88–89 (b) Query 151–200 on AP88–89

Fig. 7. Effects of Multiple Windows

Remark 1: We also test the effect of the mixture model (linear combination) of information flows and association rules. In all runs, the mixture model performs slightly better than the use of IF and AR individually (consistently by around +5% over the *IF* and less than +1% over the *AR*). This suggests that combining the different types of term relationships does not have to significantly improve retrieval performance. It also further explains the consistently better performance of association rule mining (AR) over information flow (IF). AR apparently produces better coverage of useful relationships. Therefore, using AR alone would seem to provide the basis of an effective solution.

Remark 2: The elapsed time for building the query model using AR and IF are also roughly compared. The AR (less than 1 second per query) is about three times faster than the IF (3.1 seconds per query). This further verifies our discussion about the computational issue in Section 1.

6 Conclusions and Future Work

We have proposed a novel approach, which integrates association rule mining, multiple window segmentation and query decomposition, to derive "higher-order" term relationships for query language modeling. Our framework takes into account inferences from any subset of query terms, facilitated by automatically derived multiple levels of "higher-order" term associations from the pseudo relevance feedback documents that are segmented into multiple sized chunks. A substantial suite of experiments have been conducted on various TREC collections and our approach outperforms a baseline language model, the Relevance Model and the Information Flow model. Based on experimental results, we can draw the following conclusions:

- The approach used in our paper considers the contributions from the different combinations of query words, i.e., the Q_j in Equation 4. This demonstrates that the incorporation of query decomposition to take into account all possible and partial inferences from the query is beneficial to retrieval performance.
- The multiple length document segmentation with overlapping sliding windows brings its benefits of avoiding the problem of data sparseness and the missing useful associations.
- The use of high-order terms relationships derived via association rule mining from segmented documents has proved more effective and efficient than the Information Flow model. This is, in our opinion, a significant step forward for developing operational query language models.

In the future, we will investigate more effective weighting function $P(Q_j|Q)$ to generate further improvement. The consideration of the importance of documents in high-order term relationship discovery could also be useful to improve the effectiveness of our system. In addition, other types of term relationships such as those from the WordNet, will be incorporated. Finally, some state-of-the-art algorithms, such as Metzler's MRF based method in [12], will be explored and compared in our future work.

Acknowledgement

This work is funded by the UK's Engineering Physical Sciences Research Council (EPSRC), grant number EP/E002145/1

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large database. In *Proceedings of SIGMOD'93*, pages 207–216, 1993.
2. J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of SIGIR'03*, pages 314–321, Canada, August 2003.
3. J. Bai, D. Song, P. Bruza, J.-Y. Nie, and G. Cao. Query expansion using term relationships language models for information retrieval. In *Proceedings of CIKM'05*, Bremen, Germany, November 2005.
4. N. Balasubramanian, J. Allan, and W. B. Croft. A comparison of sentence retrieval techniques. In *Proceedings of SIGIR Poster Sessions*, Amsterdam, July 2007.
5. G. Cao, J. Nie, and J. Bai. Integrating term relationships into language models. In *Proceedings of SIGIR'05*, pages 298–305, 2005.
6. J. Gao, J. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *Proceedings of SIGIR'04*, pages 170–177.
7. M. Hahsler, C. Buchta, and K. Hornik. Selective association rule generation. *Computational Statistics*, 23, 2008.
8. J. Hipp, G. U., and N. G. Algorithms for association rule mining - a general survey and comparison. In *SIGKDD Exploration*, pages 1–58, 2000.
9. J. Lafferty and C. Zhai. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *Proceedings of SIGIR'01*, pages 111–119. ACM Press, 2001.
10. V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of SIGIR'01*, pages 120–127, New York, 2001.
11. X. Liu and W. B. Croft. Passage retrieval based on language models. In *Proceedings of CIKM'02*, November 2002.
12. D. Metzler and W. B. Croft. Latent concept expansion using markov random fields. In *Proceedings of SIGIR'07*, July 2007.
13. J. Pickens and A. MacFarlane. Term context models for information retrieval. In *Proceedings of CIKM'06*, pages 559–566, November 2006.
14. A. R., I. T., and S. A. Mining association rules between sets of items in large databases. In *Proceedings of the SIGMOD'93*, pages 207–216, Washington D.C., May 1993.
15. D. Song and P. D. Bruza. Towards context sensitive information inference. *Journal of the American Society for Information Science and Tecnology*, 54(3):321–334, 2003.
16. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'01*, pages 334–342.