Query Expansion using Association Matrix for Improved Information Retrieval Performance

Jedsada Chartree[1], Ebru Celikel Cankaya[2], and Santi Phithakkitnukoon[3]
[1]Department of Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA
[2]Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA
[3]Computing Department, The Open University, Milton Keynes, United Kingdom

Abstract— *We propose a novel query expansion technique that employs association matrix to solve the problem of false positives: retrieving irrelevant documents, while missing actually required documents in a typical search engine environment. We present underlying infrastructure of our design, together with comparisons with existing query expansion algorithms and University of North Texas (UNT) Google search engine. Our results yield 14.3% improved Information Retrieval (IR) performance with more effective and precise retrievals than a conventional (non-expanded) search engine.*

Keywords: Query expansion, association matrix, information retrieval, relevance, mismatch, precision

1. Introduction

With the rapid growth of Internet technology, the number of online users is constantly on the rise, and so are their operations. *Information Retrieval (IR)*, being one of the most common operations that is used frequently by Internet users, may cause two problems: The search engine may retrieve *irrelevant* documents, and/or more importantly it may miss the *relevant* documents. These are the fundamental reasons why we get IR failures frequently. This paper proposes query expansion technique that employs association matrix to solve these two problems.

In recent years, the information content on the *World Wide Web* has been increasing in an amazing rate. This content overload introduces new challenges to the process of IR, such as delayed retrieval time, poor precision and recall rates, obscurity in word sense disambiguation, and difficulty in relevance feedback for the search engine [8, 13].

The essential problem with information retrieval is word mismatch, which typically occurs when users submit short and ambiguous queries. These queries most of the time result in retrieval of irrelevant documents, while missing the actually required (*relevant*) documents [8, 12, 13, 16]. To overcome this problem, a technique called *query expansion (QE)* has been proposed and is being widely used. The idea behind query expansion is to first add terms with close meaning to the original query to expand it, then reformulate the ranked documents to improve the relevant performance of the overall retrieval [6, 8, 11].

In this work, we improve the query expansion technique by integrating the *association matrix* concept. With this simple and fast expansion, we obtain better retrieval rates. We compare the relevance feedback results of the non-expanded (original) query implementation with that of the query expansion technique we propose by running two schemes separately. Moreover, we compare these two results with that of *UNT Google search engine*l.

The remainder of the paper is organized as follows. Section 2 reviews the background and motivation. Section 3 introduces the scheme we propose by explaining the search engine infrastructure that uses *Vector Space Model (VSM)* and query expansion with association matrix. We also describe the experimental setup in Section 3. Section 4 presents the results. The paper is concluded with a summary and an outlook on future work in Section 5.

2. Background and Motivation

Query formulation is one of the most important tasks, which has a direct impact on the relevance feedback rate in *Information Retrieval (IR)* systems [6, 8]. Many query expansion techniques have been proposed to improve the search performance, which are measured with *relevance feedback* and *pseudo-relevance feedback* values.

To obtain better retrieval rates, one can try expanding the original query. The majority of earlier work on query expansion concentrate on exploiting the term co-occurrences within documents. Unfortunately, most of the time queries are short, rendering this method inadequate. To improve this naive idea of query expansion, Gao et al. [9] propose expanding queries by mining user logs, namely by utilizing user interactions that are recorded in user logs. By analyzing the user logs, authors extract correlations between query terms and document terms. These correlations are then used to select high-quality expansion terms for new queries. Their method yields outperforming results over the current conventional search methods.

In [12], Li et al. propose a new approach to query expansion by combining thesauri and automatic relevance feedback methods. Using thesauri for query expansion is a very straightforward implementation: given a user query, the system performs a simple table look up for related terms

from thesaurus and performs expansion accordingly. This technique comes with its obvious drawbacks: Most of the time, thesauri are built manually and hence they suffer from being too broad or on the contrary too concise. Moreover, building a thesaurus involves a thorough knowledge base analysis, which may get impractically slow, especially when dynamic updates are required frequently. And finally, human interaction in the preparation of the thesaurus makes it far from objective at most times. User feedback, as the name implies, is a cyclic feedback [2] that is obtained from actual users of the system, in the hope that they will help improve future retrieval efforts. Most of the time, user feedback may not perform well, due to the high rate of subjectivity involved in it. For this reason, a better approach called *automatic relevance feedback* is adopted. This technique eliminates the human factor by assuming that the top *n* retrievals are the most relevant ones. Then, by using statistics, additional terms are selected from these *n* documents. It is this statistical processing that makes automatic relevance feedback approach too complex and too slow to implement. By bringing together thesauri and automatic relevance feedback techniques, Li et al. shows better performance over traditional methods. Nonetheless, their implementation requires complex initial setup, and may involve significant latency in retrieval time.

Mining user logs for query expansion purposes is another common technique that is referred to by many scholar work. In [13], Peng and Ma expands this idea: They propose a theme-based query expansion scheme that extracts user intent through click-through data that is available on Web sites. This technique takes advantage of the classical search methods, e.g. Vector Space Model (VSM), and adds more features to them, such as close meaning terms and synonymous words.

Yue et al. propose using text classification as a means to obtain better query expansion in [17]. They first use text classification to classify the obtained document collection, then extract key phrases from each document head to eventually build a key phrase set. Their work yields promising precision against recall values with high retrieval rates. In our work, by introducing association matrix to the conventional Vector Space Model, we are able to get comparable rates by obtaining improved performance and reduced non-relevance feedback results.

In literature, there have been a number of works on query expansion that is applied to different source languages. For example, [10] designs and develops the query expansion scheme for answering document retrieval in the Chinese answering system. Their method extracts related words and expanded the query for a specific question. The study used the Vector Space Model and cosine similarity techniques. Their results show promising relevance feedback. There is, however, no performance comparisons with other existing techniques. In another work, Gao et al. [9] suggest techniques for cross-lingual query: retrieving results in languages other than the submitted query.

In another linguistic implementation of query expansion, Kannan et al. [11] use a different source language. They present a comparison between interactive and automatic query expansion techniques applied on Arabic language. According to their results, the automatic query expansion method gives much better relevance feedback than non-query expansion. In our work, we use English as the source language to evaluate our framework. This provides us the flexibility and generality in performance comparisons with similar work. Still, our work is a generic model that can be adopted by any source language.

Applying the standard technique of query expansion to data other than text retrieval is a straightforward and effective idea. As an example, Rahman et al. implement query expansion to improve image retrieval recall and precision values. In their work [14], they use Support Vector Machines (SVM) to generate a classification of images, which is similar to vocabulary classification of text.

3. Design

In this section, we describe our novel approach to query expansion, which provides a simple and fast means to achieve better information retrieval rates with higher precision. With our design, we also aim at alleviating, if not totally eliminating, the drawbacks of existing query expansion algorithms.

3.1 Search Engine Based On Vector Space Model

Vector Space Model is commonly used for finding the relevant documents as a result of a search engine query [5]. With this model, queries and documents are assumed as a vector in an *n*-dimensional space, where each dimension corresponds to separate terms or queries. The values of the vector represent the relevant documents. Basically, we compare the deviation of angles between each document and query vector. In practice, the Vector Space Model is considered as the cosine similarity between document vectors. The cosine similarity [11, 17] can be expressed as follows:

$$sim(d_i, d_k) = \frac{\sum_{i=1}^{n} w_{i,j}\, w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^{2}}\sqrt{\sum_{i=1}^{n} w_{i,k}^{2}}} \tag{1}$$

where

$$w_{ij} = tf_{ij} idf_i \tag{2}$$

$$tf_i = \frac{f_{ij}}{max\{f_{ij}\}} \tag{3}$$

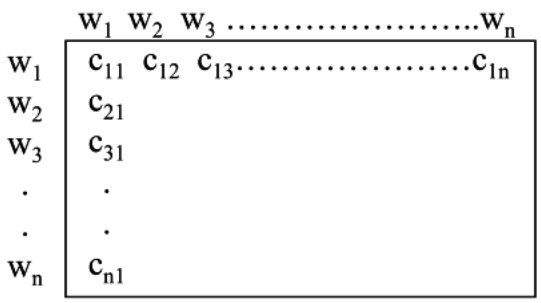

Fig. 1: Association Matrix

$$idf_i = log(\frac{N}{df_i}) \tag{4}$$

and $sim(d_i, d_k)$ is the cosine similarity between document d_i and query d_k, where w_{ij} is *tf-idf* weighting of term *i* in document *j*, tf_{ij} is term frequency of term *i* in document *j*, f_{ij} is frequency of term *i* in document *j*, idf_i is the inverse document frequency of term *I*, *N* is the total number of documents, and df_i is document frequency of term *i*.

3.2 Query Expansion Based On Association Matrix

In our proposal for the query expansion scheme, we combine two approaches to achieve better information retrieval performance: query expansion and association matrix. Query expansion is a technique that is used to expand a short query in order to obtain more relevant and more precise documents as a result of querying the search engine [8, 15, 17]. The expansion is achieved by adding other terms that are related to the original queries [6]. After query expansion, instead of the original query, the new expanded query is used in the Vector Space Model.

Association matrix is a 2-dimensional matrix, where each cell c_{ij} represents the correlation factor between all terms in a query and the terms in documents (Fig. 1). This matrix is used to reformulate an original query to improve its retrieval performance [3].

Each correlation factor, denoted as c_{ij} in Fig. 1, is calculated as follows:

$$c_{ij} = \sum_{d_k \in D} f_{ik} \times f_{jk}, \tag{5}$$

where c_{ij} is the correlation factor between term *i* and term *j*, and f_{ik} is the frequency of term *i* in document *k*. Additionally, these correlation values are used to calculate the normalized association matrix [9] as follows:

$$s_{ij} = \frac{c_{ij}}{c_{ii} + c_{jj} - c_{ij}}, \tag{6}$$

where s_{ij} denotes normalized association score, and c_{ij} represents the correlation factor between term *i* and term *j*.

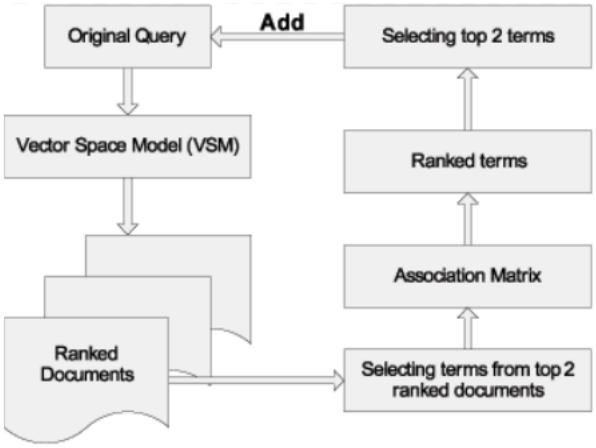

Fig. 2: The Association Matrix Query Expansion and Retrieval Framework

Higher normalized association score implies higher degree in correspondence with the original query. Thus, we choose several words, which have the highest association score, to add into the original query, then use this new query to calculate the cosine similarity instead of the original query.

3.3 Document Ranking

Fig. 2 illustrates the framework for our implementation: it brings together two main components for our design – the Vector Space Model and the association matrix. The Vector Space Model is used to re-represent a text document by applying the sequence of procedures as follows: Document indexing that is achieved by filtering out function words, etc., term weighting, and ranking the document with respect to the query according to a similarity measure. The association matrix is the 2-dimensional matrix that was explained in Section 3.2.

In the following subsections, implementation details for each component in Fig. 2 are described in more depth.

3.3.1 The Term-Weight Document

The term-weight document was used to calculate the cosine similarity value. It was generated by the following steps:

i. Crawling Web pages module: The crawling module of our program crawls 3000 Web pages of the University of North Texas (UNT) using *Breadth-First Search (BFS)* approach. This many number of web pages are good enough to be considered as part of a corpus system.

ii. Preprocessing module: This module is a combination of several tasks which includes removing SGML tags, tokenizing each word, eliminating stop words, and stemming each word using Porter Stemmer [7] to make it a root word.

iii. Indexing module: The indexing module calculates the term-weight of each word using Eq. (2). The results are stored as the term-weight document (text file), which was

Table 1: Example of Precision Values Obtained Query Expansion of the term "*career*" in different top pages and different top words

	Precision								
#Pages / *#Words*	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
2	0.971	0.783	0.745	0.403	0.413	0.392	0.41	0.553	0.404
3	0.968	0.783	0.734	0.598	0.413	0.597	0.41	0.584	0.553
4	0.971	0.78	0.734	0.607	0.413	0.587	0.41	0.583	0.403
5	0.953	0.78	0.732	0.789	0.413	0.743	0.41	0.446	0.405
6	0.886	0.852	0.796	0.422	0.772	0.75	0.489	0.446	0.407
7	0.885	0.468	0.432	0.777	0.776	0.423	0.41	0.406	0.401
8	0.573	0.468	0.436	0.422	0.412	0.412	0.409	0.405	0.406
9	0.574	0.466	0.436	0.422	0.412	0.412	0.409	0.406	0.406
10	0.574	0.468	0.436	0.422	0.429	0.412	0.239	0.352	0.301

a combination of the Web pages (URLs), words, and *tf-idf* value.

3.3.2 The Web Interface Search Engine

The Web interface search engine module uses the Common Gateway Interface (CGI) protocol to achieve the following tasks:

1) Preprocess the query.
2) Calculate the cosine similarity between the query and the documents using the term-weight values in the term-weight document file that was built in Section 3.3.1.
3) Rank documents in descending order, based on the higher cosine similarity value from the previous step, and display the results on the Web browser.

3.3.3 The Query Expansion

This step is the intelligent part of the search engine that applies an association matrix algorithm to the search engine. To accomplish the desired query expansion, several tasks are executed as follows:

1) Choose the words from top two pages described by step 3 in Section. 3.3.2. to calculate the correlation factor between these words and the original query; this step uses Eq. (5).
2) Use the result from previous step to calculate the normalized association score with Eq. (6).
3) Rank these terms in descending order, based on the normalized association score.
4) Select the top two words from previous step as an expanding query and adding these words to the original query.
5) Use the new combination of these terms to calculate the cosine similarity as described in step 2 of Section 3.3.2.
6) Rank the documents as described in step 3 of Section 3.3.2 and display the results on the Web browser.

As an example, Table 1 shows precision values of the term "*career*" in different top pages and different top words.

4. Results

We first present the results of query expansion task from subsection 3.3.3 above as in Fig. 3, which shows that the pairs (*top 2 words, top 2 pages*) yield the highest average precision of ten sample queries. Note that while constructing the query expansion, choosing the number of pages (step 1 above) and selecting the number of words (step 4 above) directly affect the overall performance of our scheme due to the larger number of pages and the larger number of words will result in more irrelevant documents. Therefore, only the top two expansion terms from the top two pages are used to combine with the original query. We then compare the performance of our method (new query after expansion) to other two search engines: the original (without expansion) and UNT Google search engines. Figure 4 and Table 3 show the results from using ten sample queries in Table 2.

According to Fig. 4, the query expansion technique either improves or maintains the current retrieval performance, with

Table 2: Query terms of the ten user's queries

User's queries	Query terms
Q1	faculty
Q2	career
Q3	engineering
Q4	gerontology
Q5	computer lab
Q6	mikler
Q7	ieli
Q8	admission
Q9	orientation
Q10	discovery

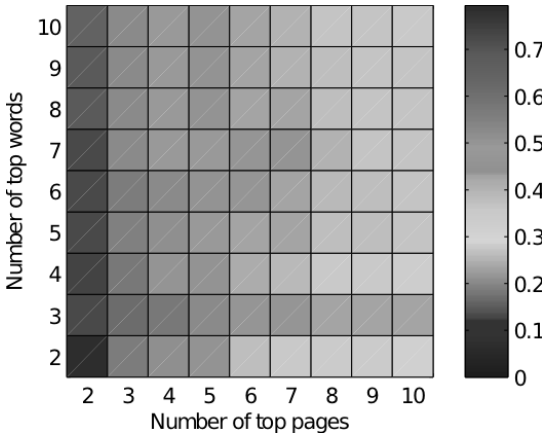

Fig. 3: Average Precision of 10 Sample Queries with the Number of Top Pages and Words Varying From 2 to 10.

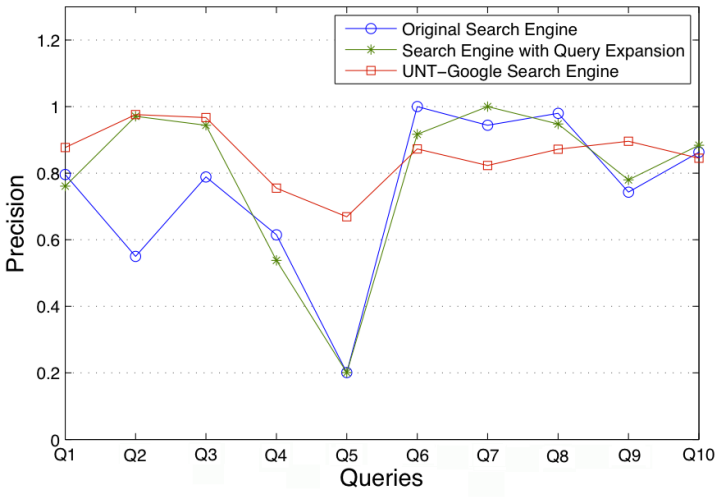

Fig. 4: The Comparison of Precision between Original Query, Query Expansion, and UNT Google Search Engine

the exception of query no. 1, 4, 6, and 8, but the precision values of these queries are very close the each corresponding query (between the original query and expansion query). This exception reminds us the fact that query expansion does not always guarantee better retrieval rates. Sometimes, it may even be the case that the unexpanded query is a better fit than its expanded counterpart.

In addition, the precision values of both the original and proposed search engines are less than the precision value of UNT Google search engine except the values of queries numbered 6, 7 ,8, and 10; this implies that sometimes the UNT Google search engine not always a better fit than the other two methods.

Overall, our proposed method (query expansion) improves the search engine's performance, particularly, it has better relevant feedback than the original search engine (without query expansion). Table 3 shows the average precision rate of 0.794 with query expansion and 0.748 without expansion. This indicates a 14.3% improvement on average. The UNT Google search however appears to outperform our proposed method with an average precision rate of 0.855.

Table 3: A Comparison of the Average Precision Rate of the Proposed Method with the Original and UNT Google Search Engines

Methods	Average Precision Rate
Original Search Engine	0.748
UNT Google Search Engine	0.855
Proposed Search Engine with Query Expansion	0.794

To better explain the discrepancies in Fig. 4, we analyze each query in more detail. According to this analysis, we observe that query no.5 ("*computer lab*"), has a relatively low score. This may suggest that both of our search engines are not suitable for the queries that have more than one word; in fact, the search engines search the query that contains more than one word separately (as *computer* and *lab*), and there are many relevant feedbacks for both of them. This presumably causes the undesired low precision rate. Nevertheless, some other queries, such as queries numbered 6, 7, and 8, of both search engines (original search engine and search engine with query expansion) have higher precision values than the UNT Google search engine. This implies that both search engines work well for a person's first name (or last name) and abbreviation (*ieli* stands for Intensive English Language Institute). Especially for expanded query like query no. 6 ($mikler$) – it returns a new query contained both the first name and last name of a faculty member of UNT.

Furthermore, between the original queries and expanded queries, we notice that the expanded queries mostly return higher precision than unexpanded queries, and return relevant document differently. In fact, the ranked documents listed after applying query expansion are more relevant documents and are ranked higher towards the top of the ranking list as shown in Fig.5.

5. Conclusion and Future Work

The rapid growth of World Wild Web makes it more and more challenging for information retrieval to achieve desired performances, especially in obtaining relevant feedback for short queries. This work implements a query expansion by using association matrix to improve the retrieval performance. Our experimental results show a 14.3% performance improvement on average. Therefore, the results of these experiments indicate that query expansion using association matrix is provably efficient in improving the ranked relevant feedback of documents. Although our proposed search engine's performance is still slightly lower than the UNT

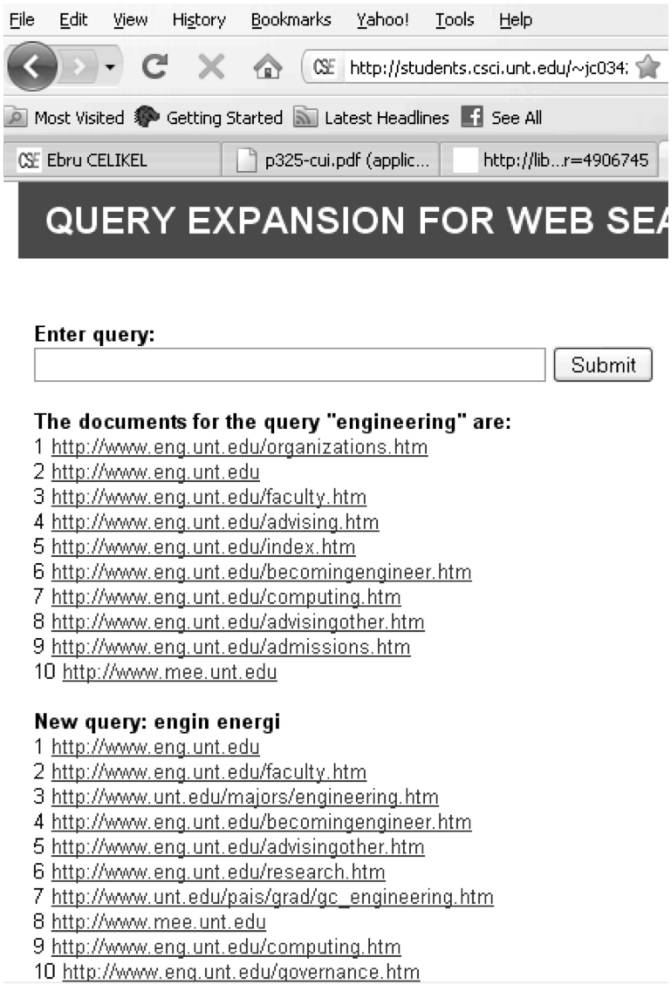

Fig. 5: The Results of Ranked Documents with query 6: "*engineer*"

Google search engine, it is not significant. This probably has to do with the different corpus that we use, i.e., for our search engines, we crawl only 3,000 webpages, and the UNT Google search engine use another corpus and another algorithm.

As our future work, we will continue to investigate the use of association matrix, as well as other techniques such as employing pseudo relevance feedback (PRF) [2, 4], or using genetic algorithm [1, 15] to improve the query expansion. An extensive comparison of these techniques will also be explored and studied in the future.

Moreover, we are planning to implement our scheme on different source languages to investigate how linguistic characteristics influence the performance of our scheme.

References

[1] L. Araujo and J. R. Pï£¡rez-Aguera J. R., "Improving query expansion with stemming terms: a new genetic algorithm approach," in Proc. EvoCOP'08, 2008, pp. 182-193.

[2] K. Belhajjame, N. W. Paton, S. M. Embury, A. A. A. Fernandes, and C. Hedeler, "Feedback-based annotation, selection and refinement of schema mappings for dataspaces," in Proc. EDBT '10, 2010, pp. 573-584.

[3] A. M. Boutari, C. Carpineto, R. Nicolussi, "Evaluating term concept association measures for short text expansion two case studies of classification and clustering," in Proc. EDBT '10, 2010, pp. 163-174.

[4] M. Cartright, J. Allan, V. Lavrenko, A. McGregor, "Fast query expansion using approximations of relevance models," in Proc. CIKM '10, 2010, pp. 1573-1576.

[5] P. A. Chew, B. W. Bader, S. Helmreich, A. Abdelali, S. J. Verzi, "An information-theoretic, vector-space-model approach to cross-language information retrieval," Cambridge Natural Language Engineering, Vol. 17, pp. 37-70, Jan. 2011.

[6] P. D. Meo, G. Quattrone, and D. Ursino, "A query expansion and user profile enrichment approach to improve the performance of recommender systems operating on a Folkson," User Modeling and User-Adapted Interaction, 2010, pp. 41-86.

[7] F. N. Flores, V. P. Moreira, C. A. Heuser, "Assessing the impact of stemming accuracy on information retrieval," in Proc. PROPOR'10, 2010, pp. 11-20.

[8] L. Gan, S. Wang, M. Wang, Z. Xie, L. Zhang, and Z. Shu, "Query expansion based on concept clique for Markov network information retrieval model," in Proc. FSKD '08, 2008, pp. 29-33.

[9] W. Gao, C. Niu, J. Nie, M. Zhou, K. Wong, and H. Hon, "Exploiting query logs for cross-lingual query suggestions," ACM Transactions on Information Systems (TOIS), Vol. 28, May 2010.

[10] K. Jia, "Query expansion based on word sense disambiguation in Chinese question answering system," Journal of Computational Information Systems, Vol. 6, pp. 181-187, Jan. 2010.

[11] G. Kannann, R. Al-Shalabi, S. Ghwanmeh, and B. Bani-Ismail, "A comparison between interactive and automatic query expansion applied on Arabic language," in Proc. IIT '07, 2007, pp. 466–470.

[12] J. Li, M. Guo, and S. Tian, "A new approach to query expansion," in Proc. Machine Learning and Cybernetics, 2005, pp. 2302-2306.

[13] V. Oliveira, G. Gomes, F. Belem, W. Brandao, J. Almeida, N. Ziviani, and M. Goncalves, "Automatic query expansion based on tag recommendation," in Proc. CIKM '12, 2012, pp. 1985-1989.

[14] M. M. Rahman, S. K. Antani, and G. R. Thoma, "A query expansion framework in image retrieval domain based on local and global analysis," Information Processing and Management, vol. 47, pp. 676-691, Sep. 2011.

[15] V. Wood, "Improving query term expansion with machine learning," M. Sci. thesis, University of Otago, Dunedin, New Zealand, 2013.

[16] S. Wu "The weighted Condorcet fusion in information retrieval," Information Processing and Management, vol. 49, pp. 108-122, Jan. 2013.

[17] W. Yue, Z. Chen, X. Lue, F. Lin, and J. Liu, "Using query expansion and classification for information retrieval," in Proc. SKG '05, 2005, pp. 31-38.