A Query Expansion Based on Sentence and Vector Integration Strategy

Min Peng[1], Ming Yang[1], Songtao Sun[1], Hua Long[1], Nasir Ghani[2], Bin Ni[3]
[1]Wuhan University, [2]University of New Mexico, [3]University of South Carolina
[1]{pengm, yangming, whusunny, hlong}@whu.edu.cn, [2]nghani@ece.unm.edu, [3]nib@email.sc.edu

***Abstract*- This paper proposes a novel query expansion method to improve the average precision of the original query for information retrieval. The scheme uses a graph-based ranking algorithm to choose sentences in a manner which is different from existing sentence-based query expansion methods. At the same time a synthesis strategy for sentences is also built to construct new queries. The proposed solution is analyzed using DUC09 test collection data for update summarization task. Overall evaluation results show that the proposed method improves performance by yielding more relevant information with less noise.**

Keywords*- *Information retrival, sentence-based query exponsion, graph-based ranking algorithm.

I. INTRODUCTION

Given the rapid expansion of the global Internet, very large volumes of electronic information are now being propagated between users. Although this development provides users with a very good source of information, it is becoming increasingly difficult for them to access the information they need in the vast global data pool. Hence, finding desired information more efficiently and accurately is the key concern in information retrieval, and *query expansion* (QE) provides a viable solution approach to this problem.

Overall, QE is a popular technique for improving query performance by basically reconfiguring a base "seed" query. This method is mainly used to resolve the problem of low search precision and recall rate, i.e., caused by user input ambiguity and/or unfamiliarity of the information retrieval system environment. Hence, most QE techniques are based upon keywords and use weight adjustments of query terms and/or semantic concepts. However, recently a sentence-based query expansion method [1] has also been proposed, and this has opened up new avenues to modify original user queries.

Motivated by these methods, we propose a novel sentence-based query expansion scheme using a graph-based ranking algorithm. Specifically, we implement sentence-to-sentence, document-to-document, and sentence-to-document relations to find query-biased informative sentences within the document set. We also integrate these sentence vectors with the original query vector to generate new query expansion vectors. Our experimental results show that the proposed scheme yields improved query performance versus a baseline query strategy.

This paper is organized as follows. Section 2 gives a brief overview of previous work in the area. Section 3 then presents the proposed sentence-based query expansion scheme in detail using the graph-based ranking approach. Meanwhile Section 4 presents experiment results for a sample data set, including system comparison with *latent semantic indexing* (LSI) without QE and LSI with QE using our proposed method. Section 5 then presents conclusions and directions for future work.

II. RELATED WORK

Sentence-based QE is a new research area in information retrieval. As compared to traditional methods (i.e., such as keywords-based, semantic concept-based and adjustment of query term weight), sentence-based QE has many advantages [1]. Foremost, this method can resolve more query-biased information for QE when a sentence contains more relevant information than a term or concept. This approach can also mitigate problems with noisy information and topic shift [1]. Finally, sentence-based QE can introduce more contexts into the query than simple term-based expansion. However, the proposed solution in [1] also has some key drawbacks. First, the selection of candidate sentences for QE does not consider the relationship among sentences, relationship among pseudo-relevant documents, or relationship between sentence and pseudo-relevant documents. In addition, QE is still done based upon terms from the selected sentences, and hence these sentences themselves are not used for the QE step.

Now a key challenge in sentence-based QE is to reasonably determine the relevant sentences. Along these lines, graph-based ranking algorithms have been used in the area of automatic text summarization to provide a reliable means of achieving sentence selection. Specifically, these methods are inspired by the PageRank [2] and manifold-ranking algorithms [3] (originally used in Internet searching schemes). Now when applied within the QE context, these graph-based ranking schemes can still make full use of the relationships among sentences in order to find the most important candidate sentences for QE. Consider some details.

Earlier, Erkan and Radev proposed the LexRank scheme [4] for generic text summarization. This work built a similarity graph where nodes represent sentences and edges represent cosine similarities between sentences. The algorithm then implemented a random walk on the graph to converge to a stationary distribution by which to rank the sentences. Meanwhile, [5] looked at query-focused summarization by taking into account the relevance of a sentence to the query. Conversely, [6] developed a manifold-ranking algorithm for query-focused summarization which made full use of both the

relationships among all the sentences in the documents *and* the relationships between the given query and the sentences. Furthermore, [7] presented a framework to model the two-level mutual reinforcement among sentences as well as documents. Specifically, the authors designed and developed a novel ranking algorithm that took into account document reinforcement during the process of sentence ranking. Finally, [8] and [9] proposed a novel and generic Co-HITS algorithm to incorporate the bipartite document-sentence graph with content information from both sides, i.e., as well as the constraints of relevance.

Overall, the above methods are already in wide use in the area of automatic text summarization. Nevertheless, their further application in the QE space has not yet been considered, and this forms the key motivation for our work. Specifically, in this paper we combine sentence-based QE methods and graph-based ranking algorithms. More importantly, we fully leverage the inherent saliencies of our proposed information retrieval model and develop a novel fusion algorithm for generating new queries based upon the chosen candidate sentences. This solution is now detailed.

III. SYSTEM OVERVIEW

The overall sentence-based QE solution is shown in Figure 1 and is now detailed. Specifically, the scheme uses graph-based ranking algorithms and implements the following steps:

1. Apply LSI model (See section III B) to rank all the documents where the original query is used.
2. Use a two-level query-biased reinforcement ranking algorithm to fetch candidate sentences for QE.
3. Integrate the candidate sentence vectors into a single sentence vector as an expanded query of the original.

Further details on these individual steps are now presented.

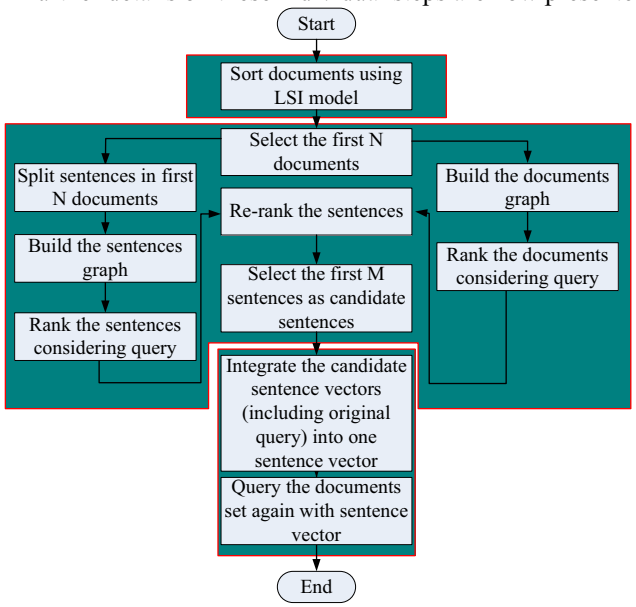

Fig. 1. System flow diagram

A. Figures and Tables

Overall, LSI is a means for data indexing and retrieval using mathematical *singular value decomposition (*SVD) techniques to identify patterns in the relationships between the terms (and concepts contained in an unstructured collection of text). In particular, LSI is based upon the principle that words used in the same contexts tend to have similar meanings. Now a key feature of LSI is its ability to extract the conceptual content in the body of a text by establishing associations between those terms that occur in similar contexts. Hence, LSI overcomes two of the most problematic constraints of Boolean keyword queries: multiple words having similar meanings (synonymy) and words having more than one meaning (polysemy). Therefore in this work we also leverage the basic LSI model to obtain pseudo-relevant documents.

B. Query-biased reinforcement ranking algorithm

A query-biased reinforcement ranking algorithm is used to fetch candidate sentences. This algorithm is intuitively based upon the following assumptions.

> **Assumption 1:** A sentence should be significant if it is heavily-linked with other significant sentences related to the query. A document should be significant if it is heavily- linked with other significant documents related to the query.
>
> **Assumption 2:** A sentence should be significant if it is in a significant document. A document should be significant if it contains more significant sentences.

Based upon the above, we develop a novel two-layer graph model to fuse three kinds of relationships, i.e., between sentences, between documents, and between sentences and documents). Specifically, consider an undirected sentences graph model denoted by $G_{SS}=<V_S, E_S>$, where $V_S = \{S_i \mid 1 \le i \le \sum_{j=1}^{N} S\_No(D_j)\}$ is the set of sentence vertexes in pseudo-relevant documents, $S\_No(D_j)$ is the number of sentences in D_j, and $E_S = \{E_{S_i,S_j} \mid sim_{LSI}(S_i,S_j) > \varepsilon\}$ is the set of edges between vertices when the LSI model similarity exceeds a threshold ε. Using this, the undirected documents graph model is denoted as $G_{DD} =< V_D, E_D >$, where $V_D = \{D_i \mid 1 \le i \le N, First\_N(sim_{LSI}(q_o, D))\}$ is the set of document vertices whose similarity score with the original query is in the top N and $E_D = \{E_{D_i,D_j} \mid sim_{LSI}(D_i,D_j) > \eta\}$ is the set of edges between vertexes when the LSI model similarity score exceeds a threshold η.

Now in these sub-graphs, each vertex has an initial weight which is equal to the relevance score between the sentence and the original query (or the document and the original query in the LSI case). Specifically, w_{si} represents the weight of the *i*-th sentence vertex and w_{di} represents the weight of the *i*-th document vertex. Also, the edges between sentences and documents denote subordinate relationships. This proposed graph model is further illustrated in Figure 2.

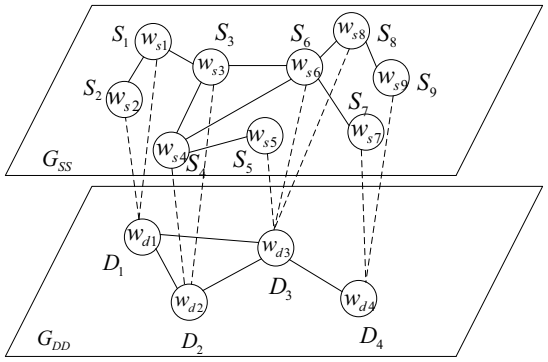

Fig. 2. Graph model of the system

The overall proposed query-biased reinforcement ranking algorithm can be divided into three sub-steps:

- Update the initial weights of the sentence vertexes based upon the sentence graph.
- Update the initial weights of the document vertexes based upon the document graph.
- Use an iterative algorithm to reinforce the first two update results.

Now the first two steps in the above graph-based ranking algorithm are similar to the PageRank scheme. Namely, the basic idea of this scheme is to determine the score of a sentence or a document by the relevance (of the sentence or document) with respect to both the query and other sentences or documents. Hence, in this way the importance of sentences or documents can be "spread" to nearby neighbors via the initial weights of the vertex associated with the query relevance. Furthermore, the process is iterated until a stable state is achieved and then all sentences are ranked according to their final scores to generate candidate sentences.

More formally, this process can be defined as follows. Let *Pr*(*j*) represent the score of a sentence or document, determined as the sum of its similarity with the query and the similarities with the other sentences or documents in the pseudo-relevant document set, i.e., denoted by Eqs. (1) and (2):

$$Pr(i) = (1-d) + d\sum_{j=1}^{n} w_{vertex}(j) \cdot \frac{Pr(j)}{O_i} \quad (1)$$

$$w_{vertex}(j) = sim_{LSI}(v_j, v_q) \quad (2)$$

where d is a damping factor from [0,1], O_i is the degree of vertex *i*, and $sim_{LSI}(v_j, v_q)$ is the similarity between the vector of vertex *j* and the vector of the original query based upon the LSI model. Using this, the scoring process for each vertex is detailed in the pseudo-code listing in Algorithm 1.

Now even though the above step generates weights (for documents and sentences), further considerations are needed to handle mutual constraints and influences between documents (sentences). Along these lines, the rest of this section introduces a reinforcement ranking algorithm on the results. Namely, the mutual reinforcement framework of [7] is leveraged here, although we only consider the relationships between document nodes and sentence nodes, i.e., and not sentence-to-sentence and document-to-document relationships (which are considered in Algorithm 1). In line with the earlier assumption, the pseudo-code for the mutual reinforcement algorithm is detailed in Algorithm 2.

Algorithm 1: S/D_Query_rank (*graph*, *damping_factor*, *max_iterations*, *min_delta*)

Input: *graph* (sentences graph or documents graph). *damping_factor*, *max_iterations*, *min_delta*.

Output: The score of each vertex in graph.

```
1: graph_size←the number of nodes in graph
2: min_value ←(1.0-damping_factor)/graph_size
3: vertexrank←Init the nodes with weight of 1.0/graph_size
4: count←0
5: For i in range of max_iterations
6:     diff←0; //total difference compared to last interaction
7:     count←count+1;
8:     For each node in graph:
9:         rank←min_value;
10:        For adjacent nodes of each node in graph:
11:            rank←rank + (weight of node×damping_factor×
adjacent nodes' vertexrank)/the number of adjacent nodes;
12:        End;
13:        diff←diff+| vertexrank [node] – rank|;
14:        vertexrank [node]←rank;
15:    End;
16:    If diff < min_delta:
17: End;
18: return vertexrank;
```

Algorithm 2: m_reinforce (*S_scores*, *D_scores*, *lambdas*, *lambdad*, *eta*, *imax*)

Input:

S_scores: Get from Algorithm 1 on sentences graph
D_scores: Get from Algorithm 1 on documents graph
lambdas: Sentence weight
lambdad: Document weight
eta: Stop condition of iteration
imax: Maximum number of iterations

Output:

The score of each sentence in S_D_graph.

```
1: For each weight of sentence node (w_d) in S_scores:
2:    For each weight of document node (w_s) in D_scores:
3:       if sentence in document:
4:          diff←1
5:          While l < imax:
6:                w_s(i)← lambdas ×w_s(j)+(1- lambdas)×
w_d(j)
7:                w_d(i)← lambdas ×w_d(j)+(1- lambdas)×
w_s(j)
8:             diff←Max(|w_s(i)-w_s(j)|,| w_d(i)- w_d(j)|)
9:             if diff < eta:
10:               break;
11:         End;
12:      End;
13: End;
14: return S_scores
```

Overall, Figures 3 illustrates some sample graph generation results on sentences and documents. Figure 4 is the rank result of first N documents which have the highest relevance scores that represent the relevance degree between an original query and candidate documents in LSI model. Now after the above-detailed mutual reinforcement step, the first M sentences which get highest rank scores in N candidate documents can be selected in descending order. We simply name this M sentences set as "*first-M*", and treat it as the candidate sentence vectors for the integration work in next step.

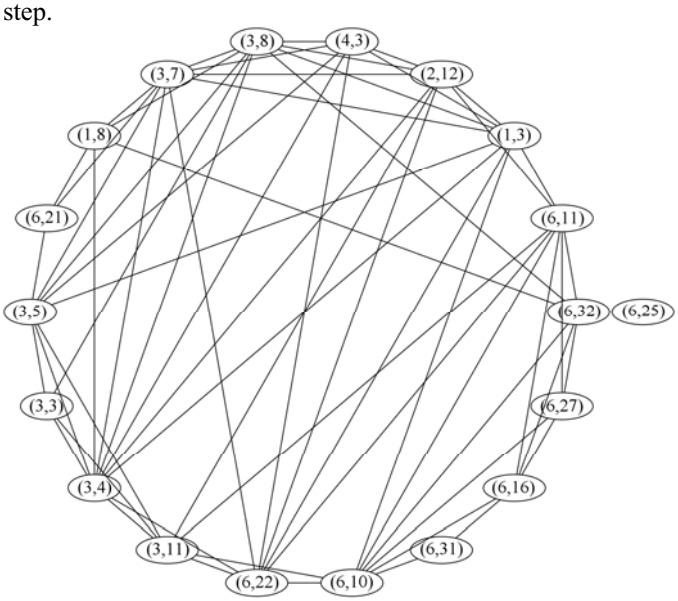

Fig. 3. A sample sentences graph (each sentence denoted by (*i*,*j*), where *i* is the document identifier and *j* is the sentence identifier)

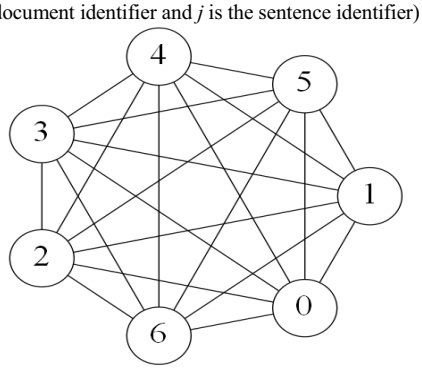

Fig. 4. A sample documents graph (each document denoted by *i*, where *i* is the document identifier)

B. Integration of the candidate sentence vectors

Now a key aim of the proposed scheme is to generate a new query based upon the original one. Hence, leveraging the candidate sentences (from Section III.B) is a major step here. Along these lines, the proposed solution introduces another key assumption as follows:

> **Assumption 3:** If more sentences belong to the same document in *first-M* (Section III.B), then that document should be treated as highly-relevant to the original query. In turn, the sentences within this document should also be considered more relevant to the query.

Based upon the above, we propose the following solution:

- For each one of the *first-N* documents, count the number of sentences containing in the *first-M* sentences set.
- Calculate the weight of each sentence vector using the following formula:

$$w_i = \frac{No_j(s)}{M}, 1 \le i \le M, 1 \le j \le N \qquad (3)$$

where s denotes the set of *first-M* sentences, $No_j(s)$ denotes the number of sentences which belong to the *j*-th document (the j-th document belongs to the set of the *first-N* documents), and w_i denotes the weight of the *i*-th sentence. In particular, w_q=1 (w_q is the weight of the original query).

- Integrate the sentence vectors and original query vector into a single vector via the following formula:

$$new\_s_k = \frac{\sum_{i=1}^{M} w_i \cdot s_{ik} + w_q \cdot q_k}{M+1}, 1 \le k \le VD \qquad (4)$$

where new_s_k is the *k*-th dimension of the final expanded query vector, s_{ik} is the *k*-th dimension of the *i*-th sentence vector, and *VD* is the number of the vector dimensions.

As per the above, a new query expansion vector is generated for use in the retrieval process.

IV. EXPERIMENTAL EVALUATION

The proposed QE scheme is analyzed using experimental data from the TAC 2009 update summarization task [11]. This data comprises of 44 document sets, with each set containing 20 documents, i.e., 880 documents in total. In particular, a document comprises of a title and narrative and the documents span across 44 topics. Now of the 20 documents in a set, a total of 15 documents are used as a training subset, i.e., for constructing the LSI model. Meanwhile, the remaining 5 documents are used as a testing subset, and query titles are selected by using the topic titles.

The performance analysis compares sentence-based QE using the proposed graph-based ranking algorithm (denoted at QE_LSI) against a baseline LSI scheme without QE (denoted as LSI). For Algorithm I, the *damping_factor* is set to 0.85, *max_iterations* is set to 100, and *min_delta* is set to 0.00001. Meanwhile for Algorithm 2, *lambdas* is set to 0.8, *lambdad* is set to 0.6, *eta* is set to 0.000001, and *imax* is set to 100.

TABLE I
QUERY PERFORMANCE OF BASELINE LSI AND PROPOSED QE SCHEME (QE_LSI)

	Avg Recall	Avg Precision	Avg F-score
LSI	0.978188731	0.812795822	0.128745096
QE_LSI	0.98900795	0.913255928	0.133956301
abs(difference)	0.010819219	0.100460106	0.005211204

First, Table 1 presents the performance of the two schemes with regards to several key metrics, including average recall, average precision, and average F-score. The results show

consistent improvements with the proposed graph ranking scheme, with almost 10% increase in precision. Meanwhile, Figure 5 also plots some of these metrics for a sample query set. For example, Figure 5(a) compares the precision rate of the baseline LSI and modified QE schemes. These results indicate generally increased variability with the baseline LSI scheme, yielding lower precision as compared to the proposed scheme. Meanwhile, Figure 5(b) plots the recall rates, showing generally similar performances between the two schemes. Finally, the average F-scores are plotted in Figure 5(c), and show notably lower variability (and higher scores) with the modified QE scheme. These results confirm that the proposed scheme can effectively improve query performance.

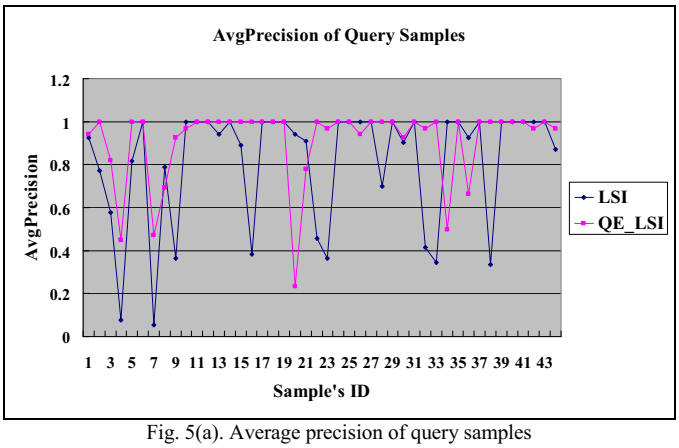

Fig. 5(a). Average precision of query samples

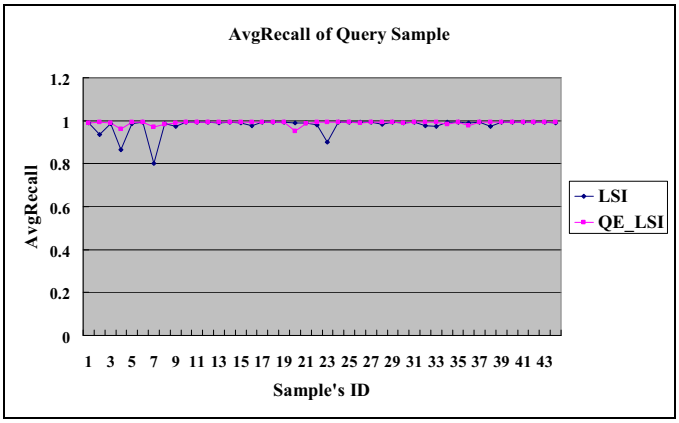

Fig. 5(b). Average recall of query samples

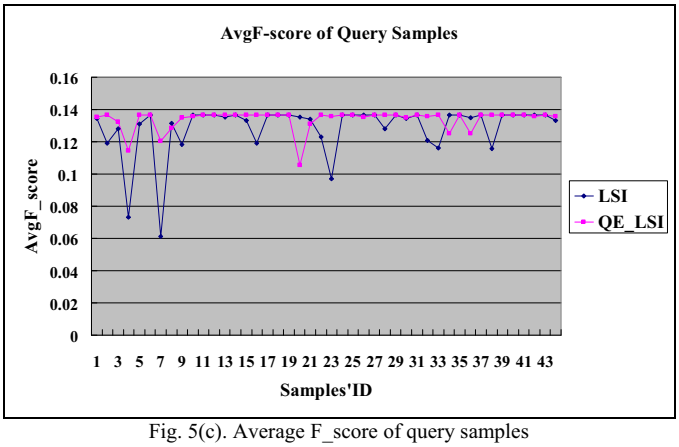

Fig. 5(c). Average F_score of query samples

Carefully note that the above findings also indicate that the proposed scheme can yield lower precision for some of the queried terms, i.e., see Figure 5(a). This indicates that the stability of the scheme needs to be improved, and this is the focus of some ongoing efforts.

V. CONCLUSIONS

This paper presents a novel sentence-based query expansion scheme using a graph-based ranking approach. Namely, this solution considers three relationships to select sentences from pseudo-relevant documents generated by the original query, i.e., sentence-to-sentence, document-to-document, and document-to-sentence. In addition, a fusion algorithm is also introduced to integrate candidate sentences and original query terms in a latent semantic indexing framework. The proposed scheme is analyzed using a realistic experimental dataset and the results show improved performance with regards to recall and precision metrics. Along these lines, future work will focus on designing a more reasonable sentence selection algorithms and also trying to improve the sentence fusion strategy.

ACKNOWLEDGMENT

Our research has been supported by the National Science Foundation of China (NSFC) under Award 61070083. The authors are very grateful for this generous support.

REFERENCES

[1] D. Ganguly, J. Leveling, G. Jones. G, "Query Expansion for Language Modeling Using Sentence Similarities", *Multidisciplinary Information Retrieval*, (62-77) 2011.

[2] L. Page, S. Brin, R. Motwani, T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", *Technical Report*, Stanford University, Stanford, CA, 1998.

[3] D. Zhou, "Ranking on Data manifolds", *Proceedings of NIPS 2003*, Whistler, Canada, December 2003.

[4] G. Erkan, D. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization", *Journal of Artificial Intelligence Research*, Vol. 22, 2004, pp. 457–479.

[5] J. Otterbacher, G. Erkan, Dr. Radev, "Using Random Walks for Question-Focused Sentence Retrieval", *Proceedings of HLT-EMNLP 2005*, Vancouver, BC, 2005.

[6] X. Wan, J. Yang, J. Xiao, "Manifold-Ranking Based Topic-Focused Multi-Document Summarization", *Proceedings of IJCAI 2007* Hyderabad, India, January 2007.

[7] F. Wei, W. Li, Q. Lu, Y. He, "Applying Two-Level Reinforcement Ranking in Query-Oriented Multidocument Summarization", *Journal of the American Society for Information Science and Technology*, Vol. 60, No. 10, 2009, pp. 2119–2131.

[8] H. Deng, M. Lyu, I. King, "A Generalized Co-HITS Algorithm and Its Application to Bipartite Graphs", KDD 2009, Paris, France, June 2009.

[9] P. Hu, D. Ji, C. Teng, "Co-HITS-Ranking Based Query-Focused Multi-document Summarization", AIRS 2010, Taipei, Taiwan, Dec. 2010.

[10] C. Papadimitriou, H. Tamaki, P. Raghavan, S. Vempala, "Latent Semantic Indexing: A Probabilistic Analysis", *Journal of Computer and System Sciences*, Vol. 61, No. 2, October 2000, pp. 217-235.

[11] TAC2009 Update Summarization Task, http://www.nist.gov/tac/2009/Summarization/