Journal of Information & Computational Science 9: 17 (2012) 5391–5400
Available at http://www.joics.com

A Query Expansion Based on Sentence and Vector Integration Strategy Using Universal Gravitation ⋆

Jiajia HUANG [a], Chundian LI [a], Songtao SUN [a], Nasir GHANI [b], Min PENG [a,*]

[a] *Computer School of Wuhan University, Wuhan 430072, China*
[b] *ECE, University of New Mexico, Albuquerque, NM 87131, USA*

Abstract

This paper proposes a novel vector integration strategy for query expansion based on sentences. The strategy utilizes the concept of universal gravitation in physical domain to integrate the vectors of sentences with the modified law of universal gravitation. Vectors generation here for query expansion is based on the improvement of the previous work we have done. The proposed integration strategy is analyzed by DUC09 test collection data. Overall evaluation results show that the proposed method has a better performance on information retrieval.

Keywords: Information Retrival; Sentence-based Query Exponsion; Graph-based Ranking Algorithm

1 Introduction

With the rapid development of the Internet, a large number of electronic information is being propagated between users. Although the Internet provides us a good source of information, it is difficult for users to get the information they need in the vast amount of information. Hence, the way to retrieve the desired information more efficiently and accurately is a key matter in information retrieval, and query expansion (QE) provides a viable solution to it.

Overall, QE is a popular technique for improving query performance by reconfiguring a "seed" query. It is mainly used to resolve the problem of low precision and recall on information retrieval which caused by ambiguity of user requirement and 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. Recently, a sentence-based query expansion method [1] has been proposed. It opens up new approaches to modify original user queries.

Inspired by these methods, we use the graph-based ranking algorithm proposed in [12] (Our previous work on NISS2011) to choose the candidate sentences. Then, we integrate these candidate

⋆Project supported by the National Nature Science Foundation of China (No. 61070083).
*Corresponding author.
Email address: pengm@whu.edu.cn (Min PENG).

December 15, 2012

sentence vectors with the original query vector that utilizes the modified universal gravitation formula, which is an upgrade version of the strategy in [12]. Our experimental results show that the new strategy is better than the previous work in [12].

This paper is organized as follows. Section 2 gives a brief overview of previous work in the area. Section 3 then introduces the graph-based ranking algorithm proposed to choose the candidate sentences in [12] and presents the integration theory that on the basic of universal gravitation in detail. Meanwhile, Section 4 presents experiment results for a sample data set, including system comparison with latent semantic indexing (LSI) without QE, LSI with QE using the strategy in [12] and LSI with QE using the proposed strategy in this paper. Section 5 presents the conclusions and directions for future work.

2 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-based QE retrieval since a sentence contains more relevant information than a term or concept. This approach can also resolve the 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 candidate sentences selection 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 on terms from the selected sentences. Hence, the sentences themselves are not used for in the step of QE.

Now a key challenge in sentence-based QE is how to determine the relevant sentences reasonably. Along these lines, graph-based ranking algorithms have been applied in the area of automatic text summarization, which provides a reliable means for 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 in the QE context, these graph-based ranking schemes can still make full use of the relationships among sentences to find the most important candidate sentences in 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 represented sentences and edges represented 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 of the relevance between the sentence and 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 considered 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 as well as the constraints of relevance.

Overall, the above methods are already widely used in the area of automatic text summarization. Nevertheless, their further application in the QE space has not been considered yet, and it's the key motivation of our work. We have proposed a strategy based on the weight of respective relationships to generate the new query vector and get a better performance than the retrieval without QE in previous work [12]. But it didn't consider the distance between original query and each of the candidate sentence vectors in LSI vector space. In this paper, we introduce the concept of the universal gravitation into the integration strategy as to get more rational new query vectors for retrieval.

3 Interation Strategy Using Universal Gravitation

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

- Apply LSI model (See section III B) to rank all the documents where the original query is used.
- Use a two-level query-based reinforcement ranking algorithm to fetch candidate sentences for QE.
- By using universal gravitation, the candidate sentence vectors are integrated into a single sentence vector as an expanded query of the original.

3.1 LSI for original query

Overall, LSI is a means of data indexing and retrieval by utilizing 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 of extracting 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 have similar meanings (synonymy) and words have more than one meaning (polysemy). Therefore, we also leverage the basic LSI model to obtain pseudo-relevant documents in this work.

3.2 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.

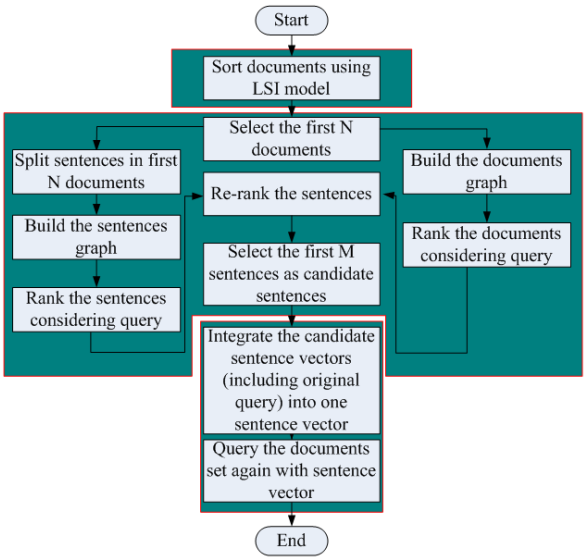

Fig. 1: System flow diagram

Based upon the above assumptions, 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 $V_s = \{S_i | 1 \leq i \leq \Sigma_{j=1}^{N} S\_No(D_j)\}$ as the set of sentence vertexes in pseudo-relevant documents, $S\_No(D_j)$ is the number of sentences in D_j, and $E_S = \{E_{s_i,s_j} | sim_{LSI}\}$ 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 $V_D = \{D_i | 1 \leq i \leq N, First\_N(sim_{LSI}(q_0, D))\}$ is the set of document vertices whose similarity score with the original query is in the top N and $E_D = \{E_{Di}, D_j | sim_{LSI}(D_i, D_j) > \eta\}$ 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). Let w_{si} represents the weight of the i-th sentence vertex and w_{di} represents the weight of the i-th document vertex. Moreover, the edges between sentences and documents denote subordinate relationships. The proposed graph model is further illustrated in Figure 2.

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/document by calculating the similarity between it and the query as well as other sentences/documents. In this way, the importance of sentences or documents can be "spread out" to nearby neighbors via the initial weights of the vertex which is associated with the query relevance. The process is

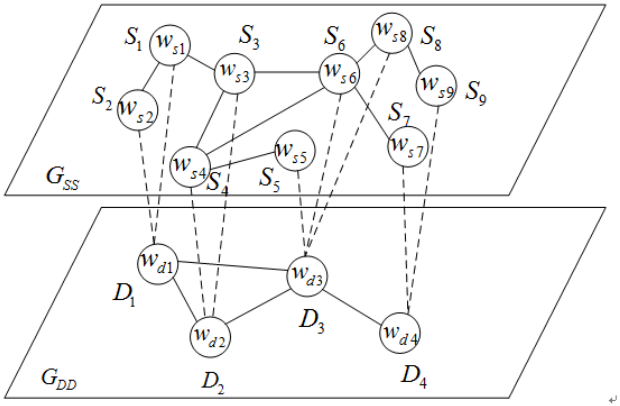

Fig. 2: Graph model of the system

iterated until a stable state is achieved. Then all sentences are ranked according to their final scores to generate candidate sentences.

More formally, this process can be defined as follows. Let $Pr(j)$ represents the score of a sentence or document. It is determined by 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):

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

$$w_{vertex}(j) = sim_{LSI}(v_j, V_q) \tag{2}$$

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.

Another thing here is, though the above steps have already generated weights for documents or sentences, further considerations are needed, to handle mutual constraints and influences between documents (sentences). Along these lines, we will introduce a reinforcement ranking algorithm in the rest of this section. To consider the relationships between document nodes and sentence nodes, the mutual reinforcement framework of [7] is utilized here (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.

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 N candidate documents set as “first-N” and M sentences set as “first-M”, and treat first-M as the candidate sentence vectors for the integration work in next step.

3.3 Integration of the candidate sentence vectors

Now a key aim of the proposed scheme is to generate a new query that based upon the original one. Hence, leveraging the candidate sentences (from Section III.B) is a major step here. Along

these lines, we will give a brief introduction of previous strategy in [12] firstly. Then the new strategy will be presented.

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: diff0; //total difference compared to last interaction
7: countcount+1;
8: For each node in graph:
9: rankmin_value;
10: For adjacent nodes of each node in graph:
11: *rankrank* + (*weight* of node×*damping_factor*×adjacent nodes' *vertexrank*)/the number of adjacent nodes;
12: End;
13: *diffdiff*+| *vertexrank* [*node*] – rank|;
14: *vertexrank* [*node*]*rank*;
15: End;
16: If *diff* < *min_delta*:
17: End;
18: return *vertexrank*;

Fig. 3: **Algorithm 1**

The previous strategy 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:

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

where s denotes the set of first-M sentences, $No_j(s_j)$ 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:

$$new\_s_k = \frac{\Sigma_{i=1}^{M} w_i \cdot s_{ik} + w_q \cdot q_k}{M+1}, \tag{4}$$

where new_{sK} 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.

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*

Fig. 4: **Algorithm 2**

As discussed above, a new query expansion vector is generated to be applied in the retrieval process.

Although the new query expansion vector has a better performance than the original one, the strategy does not have a solid theoretical support. More obviously, the strategy only considers

the effect of the relationship of affiliation between documents and sentences. It does not make full use of the conditions that LSI provided for us.

Given the defects mentioned above, we import the concept and modify the formula of universal gravitation. The classic formula of universal gravitation is as follow.

$$\overrightarrow{F}(\overrightarrow{r_i}) = -Gm_i m_0 \frac{\overrightarrow{r_i} - \overrightarrow{r_0}}{|\overrightarrow{r_i} - \overrightarrow{r_0}|^3} \tag{5}$$

We treat the candidate sentences in LSI vector space as mass points. Masses of the points are treated as query relevant degree of sentence vectors. So we modified formula (6) as follow.

$$\overrightarrow{F}(\overrightarrow{s_i}) = -\cos < \overrightarrow{s_i}, \overrightarrow{s_{org}} > \frac{\overrightarrow{s_i} - \overrightarrow{s_{org}}}{|\overrightarrow{s_i} - \overrightarrow{s_{org}}|^3} \tag{6}$$

$$\overrightarrow{F}(\overrightarrow{s_i}) = -\frac{\overrightarrow{s_i} \cdot \overrightarrow{s_{org}}}{|\overrightarrow{s_i}| \cdot |\overrightarrow{s_{org}}|} \frac{\overrightarrow{s_i} - \overrightarrow{s_{org}}}{|\overrightarrow{s_i} - \overrightarrow{s_{org}}|^3} \tag{7}$$

According to the formula (7), the total force of candidate sentences acted on original query vector is calculated as follow.

$$\overrightarrow{F} = \Sigma_{i=0}^{M} - \frac{\overrightarrow{s_i} \cdot \overrightarrow{s_{org}}}{|\overrightarrow{s_i}| \cdot |\overrightarrow{s_{org}}|} \frac{\overrightarrow{s_i} - \overrightarrow{s_{org}}}{|\overrightarrow{s_i} - \overrightarrow{s_{org}}|^3} \tag{8}$$

The $\overrightarrow{F}$ can be treated as offset measurement of the original query. So the final query expansion vector can be defined as follow.

$$s_{QE} = s_{arg} + \Sigma_{i=0}^{M} - \frac{\overrightarrow{s_i} \cdot \overrightarrow{s_{org}}}{|\overrightarrow{s_i}| \cdot |\overrightarrow{s_{org}}|} \frac{\overrightarrow{s_i} - \overrightarrow{s_{org}}}{|\overrightarrow{s_i} - \overrightarrow{s_{org}}|^3} \tag{9}$$

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

4 Experimental Evaluation

The proposed Integration strategy for QE based on sentences is analyzed by using experimental data from the TAC 2009 update summarization task [11]. The data comprises of 44 document sets, with each set containing 20 documents, i.e., 800 documents in total. More over, a document comprises of a title and narrative and the documents span across 44 topics. 15 documents in each document set are used as training set for constructing LSI model. Meanwhile, the rest 5 of documents in the document set are used as testing set, and the query samples set is build by the topic titles.

The performance analysis compares sentence-based QE by using the integration strategy with universal gravitation (denoted at QE_UG_LSI) against a baseline QE scheme proposed in [12] (denoted at QE_LSI) and LSI scheme without QE (denoted at LSI). For Algorithm 1, 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 1: QE, QE_LSI and QE_UG_LSI

	Avg Recall	Avg Precision	Avg F-score
QE	0.978188731	0.812795822	0.128745096
QE_LSI	0.98900795	0.913255928	0.133956301
QE_UG_LSI	0.989062602	0.914939196	0.134420728

First, Table 1 presents the performance of the three schemes with regards to several key metrics, including average recall, average precision, and average F-score. The results show that consistent improvements with the proposed vector integration strategy by using universal gravitation with almost 0.17% increase in precision. Meanwhile, Figure 3 also plots some of these metrics for a sample query set. For example, Figure 3(a) compares the precision rate of the baseline LSI and QE schemes by using the strategy in [12] with the QE schemes with the strategy of universal gravitation. These results indicate that generally increased variability with the baseline LSI scheme and QE schemes in [12], yielding lower precision as compared to the proposed scheme. Meanwhile, Figure 3(b) plots the recall rates, showing generally similar performances between the three schemes. Finally, the average F-scores are plotted in Figure 3(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.

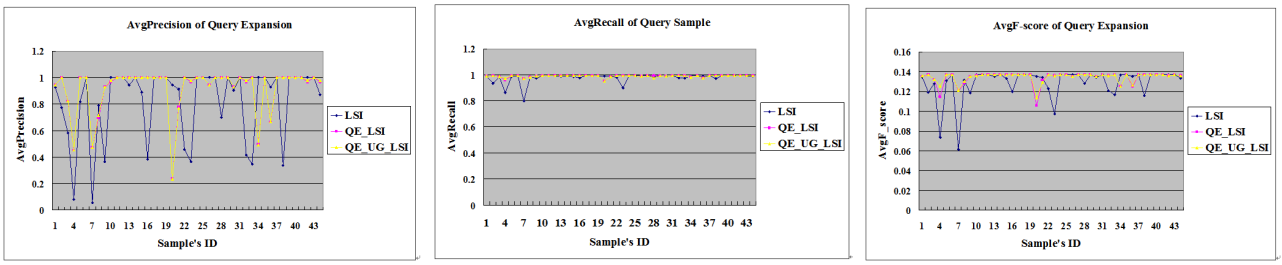

Fig. 5: (a) (b) (c) metrics for a query sample

Carefully note that the above findings also indicate that the proposed scheme can yield lower precision for some of the queried terms (as shown in Figure 3(a)). This indicates that the stability of the scheme needs to be improved, and this is the focus of some ongoing efforts.

5 Conclusions

This paper presents a vector integration strategy using universal gravitation. The strategy goes beyond the previous work we have done in [12]. The modified formula of universal gravitation is applied to integrate candidate sentences and original query terms in LSI. 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 be focused on designing a more reasonable sentence selection algorithm and improving 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, in: Multidisciplinary Information Retrieval, 2011, pp. 62 – 77.

[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, in: 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, in: 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, in: 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/.