Query Expansion based on Associated Semantic Space

Guangjun Huang
Electronic & Information Engineering College, Henan University of Science and Technology
Luoyang 471003, China
Email: guangjunhuang@126.com

Shuili Wang and Xiaoguo Zhang
Electronic & Information Engineering College, Henan University of Science and Technology
Luoyang 471003, China
Email: {wangshuili1985, zhxiaoguo}@163.com

Abstract—**Expansion of query keywords based on semantic relations is an effective approach to improve the performance of information retrieval. Traditional methods of query expansion did not adequately make use of semantic relations between query keywords. In this paper, a novel approach for query expansion is presented. The main idea of the approach is to construct a 'Tree of Associational Semantics Model' and select candidate keywords from the tree. In the first step, a group of initial semantic trees for original keywords are constructed based on WordNet thesaurus. Secondly, noise nodes on the trees are removed by calculating the similarities between words. The pruned trees are subsequently assembled into a big integrated tree, i.e. Tree of Associational Semantics Model, by expanding the trees upward until finding a common root. Finally, the nodes on the integrated tree are filtered and supplemented based on Mutual Information. All words selected from the tree are assigned semantic weights which are used in computing similarity between the query and documents in internet. In addition, the distributional situation of query keywords in documents is also considered in document retrieval. Experimental results demonstrate about 14.6% precision and 13.7% prec@20 improvement over the traditional *tfidf*-based method.**

Index Terms—**query expansion, associational semantics, weighting keywords, semantic similarity**

I. Introduction

Users usually submit only few words as their queries on the Web. It is even worse that the users' query words may be quite different to the ones used in the documents in describing the same semantics. The user may get an overwhelming but large percent of irrelevant documents in the result set. Query expansion is an effective method to solve the problems.

Query expansion means to add some synonym or relative words into the query set of original keywords to improve the recall and precision of information retrieval. Traditional methods for query expansion often get into a great trouble improving both the recall and precision. Along with the increase of the size of the query set, the recall shows rapid progress while the precision usually drops down in a way. It is caused by two main issues: the first issue is that existing approaches for query expansion just regard original keywords as a single word individually when finding relative words for it, instead of grasping the semantic of user's query on the whole and expanding the query set integrally. Another one is that existing methods do not consider semantic weights among the keywords and distribution density of keywords in documents when computing similarity between the queries and the documents.

In order to represent the semantic of query as a whole and use semantic relations of keywords to expand the set of keywords, researchers have proposed a kind of idea, namely, Conceptual Semantic Space[1][2]. The notion is that we can construct a Tree of Associational Semantics Model[TASM] firstly, and then select the candidates from TASM for query expansion. The researchers have put forward some approaches to construct a TASM, but there is not yet any maturity one having been accepted publicly in the area of information retrieval. There are still some difficult problem to the construction methods, such as using semantic similarity instead of associational semantics, depending on people's experience to assign weights to keywords, how to use the weights and distribution of keywords in the computation of similarity between the query and the document, and so on.

In order to solve the above mentioned problems, in this paper, we propose an approach to expand the set of query keywords based on associational semantics. Firstly, we construct a group of semantic trees for original keywords one by one based on WordNet, an online lexical system. The original keywords perch on the roots of the trees . Secondly, we will remove noise nodes in the trees by computing the similarity between words, and assemble the trees into a big integrated tree, i.e. Tree of Associational Semantics Model[TASM], by expanding the roots of the trees upward until finding the common origin of the trees. Using Local Analysis, we can scan the documents returned by initial retrieval, and find associated words which are not in the trees based on Mutual Information. And then, we will assign a weight to each word on the trees, and select candidates from the trees by referring to thresholds. Finally, we will execute

doi:10.4304/jcp.6.2.172-177

the document retrieval by importing the weights and distribution density of keywords into calculation of similarities between query and documents.

This paper is organized in five sections. After this introduction, a brief review of related work is provided in Section 2. Section 3 describes a new approach for query expansion. Results of experiments are presented and compared to other methods in Section 4. Section 5 brings main conclusion.

II. Related Works

Query expansion is one of the promising approaches to deal with the word mismatch problem in information retrieval. It can be roughly classified into four groups: Global Analysis[GA], Local Analysis[LA], Association Rule-based Analysis[ARA] and Query Logs-based Analysis[QLA].

The basic idea in Global Analysis is that similarities between original keywords and all words in a corpus are analyzed to get the semantic correlation (e.g. cooccurrence relationships) of every pair of words. And then, a descriptor list is constructed based on the semantic correlation. Finally, we select those words which have higher level of the correlation as candidate for query expansion. Lesk and Spark respectively used GA to expand the query set, but they did not get the result they expected[3][4]. The main reason is that GA cannot solve the polysemy problem, i.e. the ambiguity of an individual word or phrase that can be used in different contexts to express two or more different meanings. Qiu and Frei proposed a method of query expansion based on concepts in 1993[5]. Jing and Bruce put forward Phrase-finder in 1994[6]. They solved the polysemy problem to a certain extent. The common ground between these methods is that candidates should co-occur with original keywords in documents. GA is generally rather computationally demanding.

Local Analysis[LA] is the improvement of GA. LA technologies analyze only the information in some initial documents, which are retrieved for the original query. Terms are extracted from these documents for query expansion. LA can usually get higher performance than GA, but the result of LA is dominated by the initial retrieval.

Fonseca et al. used Association Rule-based Analysis[ARA] to expand his query set, and got 23.16% improvements on the precision[7]. ARA uses date mining technologies to find association rules, and selects candidates for queries from the consequent of rules. The performance of ARA depends on the quality of the date mining.

In addition, Query Logs-based Analysis[QSA] can also improve the performance of queries, but it needs a long-term accumulation of query logs. Moreover, the performance of QSA depends on whether users have a common interest in webs.

In a word, although the traditional methods for query expansion have got many improvements, they can not yet go into service reliably. The reason behind the situation is that the traditional methods just focus on single keyword to expand queries and only adopt symbol matching to search documents, instead of using semantic matching to search them. In other words, they neglect some important semantic relations between original keywords.

Zhang and Gong tried to improve traditional methods of query expansion respectively by constructing conceptual semantic space[1][2]. They selected candidates for query expansion based on semantic weights, but they did not use the weights in calculating similarities between queries and documents, which weakened the description of user's query intent. Besides, the algorithms of assigning weights in their methods are complex and depend on people's experiment too much. In addition, Aseervatham and Sujeevan presented a Concept Vector Space Model (CVSM) which uses linguistic prior knowledge to capture the meanings of the documents[8.] Liu et al. presented a method of sense recognition of hyponymy based on concept space[9]. He used the contexts of hyponymy and the weights of feature words to construct a hyponymy-word vector space. Van et al. proposed a algorithm for finding associations between related concepts, and used it in combination of information from various articles[10]. All above methods find and use semantic relations between concepts based on concept semantic architecture.

Our approach in this paper belongs to the LA category. But we combine the knowledge from the semantic dictionary WordNet with the statistic information based on a kind of mutual information technology. In addition, we adopt a different category to assign the weights of keywords and use them in calculation of similarities between queries and documents.

III. The Method Of Query Expansion

The processing flow of query expansion is shown as in the fig.1. We will introduce each process in detail in this section.

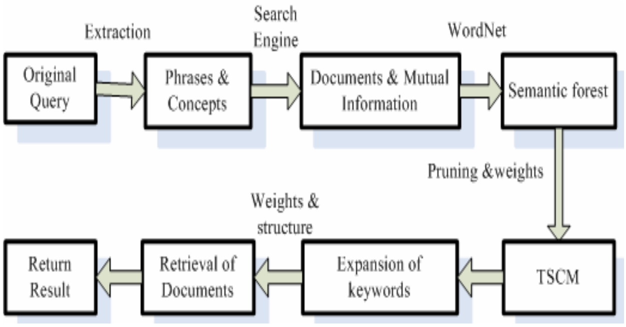

Figure 1. Flow chart of query expansion.

A. Generation of Initial Semantic Tree

There are many methods to produce and represent the semantic hiberarchy for query $Q=<q_1, q_2, \dots q_k>$. The methods can be divided into two groups. The first one is based upon statistic information and probability model to construct a conceptual semantic space for Q. For example, we can use original keywords to make a initial search in Internet, as does LA. Then, we can calculate similarities between Q and words in the documents returned by initial search based on cooccurrence frequency or mutual information, and use the similarities to build a semantic tree. This kind of methods is easy to practice, but it neglects some important semantic relations buried in natural language. Another one utilizes so-called semantic dictionary, such as WordNet. These kinds of methods have to often update the dictionary to collect the newest language knowledge. In this subsection, we will build a group of initial semantic trees based on WordNet, and then, filter and supplement information to the tree in follow-up subsections.

WordNet is an online lexical system. It contains the definitions of words and their relationships. Instead of being organized alphabetically, WordNet is organized conceptually. The basic unit in WordNet is a synonym set, or Synset, which represents a lexicalized concept.

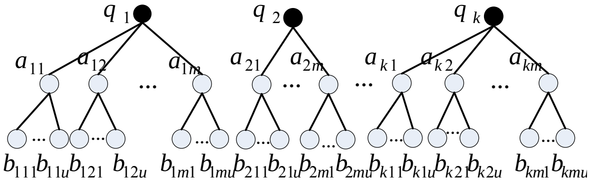
q_1
q_2
q_k
a_{11}
a_{12}
a_{1m}
a_{21}
a_{2m}
a_{k1}
a_{k2}
a_{km}
b_{111} b_{11u} b_{121} b_{12u} b_{1m1} b_{1mu} b_{211} b_{21u} b_{2m1} b_{2mu} b_{k11} b_{k1u} b_{k21} b_{k2u} b_{km1} b_{kmu}

Figure 2. Initial semantic trees

Fig.2 shows a group of semantic trees. In our trees, the original keywords of query Q are located in roots on the trees. The direct child node a_{ij} is from its hyponym in WordNet. Indirect child node b_{iju} can be got from a_{ij}'s hyponym in WordNet too. For simplicity, we only expand queries two levels along hyponymy because our goal is just to demonstrate the effectiveness of our method instead of developing a practical system. In fact, multiple hyponyms relations expand too many words which may diverge the original keyword meanings. That is, they will generate many noises in the results, thus reduce precision[2]. In practical development, we can determine the appropriate depth of the trees by observing the reaction of the precision to change of threshold values for semantic similarities.

Let $S_i=\{s_{ij} \mid 1\leqslant j\leqslant m\}$ denote q_i's synonymy set, where m is the number of the synonym, $1\leqslant i\leqslant k$. The m equals 5 in the fig.2. Let a_{ij} denote q_i's jth child node, and b_{iju} is a_{ij}'s uth child node. Now, we can build the semantic tree for query Q as follows.

Algorithm 1. Generation of initial semantic tree

```
for i = 1 to k do
call API of WordNet to get $q_i$'s synonymy set, and all
  synonyms are arrayed from 1 to m according to the
  similarities between $q_i$ and them.
  for j =1 to m do
  call API of WordNet to find $q_i$'s hyponyms, and all
    hyponyms are arrayed from 1 to m according to the
    similarities between $q_i$ and them.
    for u = 1 to m do
      call API of WordNet to find $a_{ij}$'s hyponyms, and all
      hyponyms are arrayed from 1 to m according to the
      similarities between $a_{ij}$ and them.
    end u
  end j
end i
```

B. Integrating, Pruning and Weighting

There are different similarities between words on the initial semantic trees, and some words have much less similarity than expected value. It will lead to poor precision and increase computation overhead if all these words are added to the query set. So, it is necessary to delete some noise nodes from the trees according to predetermined threshold values for similarities. As mentioned above, the semantic trees are constructed based on WordNet. In order to filter out the noise nodes and fuse more semantic information from different sources, we will use Mutual Information[MI] to prune the trees in this subsection.

In addition, there are some intrinsic semantic relations between keywords which reflect user's query intention. The intention can be represented in a way by a integrated tree which links all initial trees by expanding roots on the initial trees upward based on hypernymy. The integrated tree is called TASM.

As a preliminary treatment, we use original query Q to do initial retrieval, and get a group of resulted documents. We select top n documents as our corpus to calculate MI. A algorithm of integrating, pruning and weighting is described as follows.

Algorithm 2. Extention of initial semantic tree

```
for i = 1 to k do
call API of WordNet to get $q_i$'s hypernymy set $H_i=\{h_{ij} \mid$
  $1\leq j\leq n\}$, n is $H_i$'s maximum quantity acceptable.
if multiple original keywords share a common
  hypernym $h_c$, then $h_c$'s hypernym will be counted as
  the original keywords' one.
extend $h_c$'s hypernymy set until finding a common
  hypernym node $h$ for all roots on the initial trees.
if node $f$ is in a route, from h to $q_i$, then reserve $f$ on the
  tree; or else delete h from the tree.
end i
```

Let $sim_3(q_i, s_{ij})$ denote the similarity between original keyword q_i and its synonym s_{ij} , T_3 is the predetermined threshold for synonymy. Let $sim_1(q_i, a_{ij})$ denote the similarity between q_i and its child node a_{ij}, T_1 is the scheduled threshold for hyponymy. The $sim_1(q_i, a_{ij})$ and $sim_3(q_i, s_{ij})$ are calculated based on MI.

Algorithm 3. Pruning and weighting the integrated tree

```
for i = 1 to k do
  for j = 1 to m do
  if $sim_3(q_i, s_{ij})\geq T_3$, then assign $sim_3(q_i, s_{ij})$ to $s_{ij}$, i.e. $s_{ij}$'s
    weight is $sim_3(q_i, s_{ij})$; or else delete $s_{ij}$ and its next
    words in the $S$.
  if $sim_1(q_i, a_{ij})\geq T_1$, then assign $sim_1(q_i, a_{ij})$ to $a_{ij}$; or else
    delete $a_{ij}$ and its next words in $q_i$'s hyponymy set.
    for u =1 to m do
```

```
6:    if $sim_1(a_{ij}, b_{iju}) \geq T_2$, then assign $sim_1(a_{ij}, b_{iju})$ to $b_{iju}$;
      or else delete $b_{iju}$ and its next words in $a_{ij}$'s
      hyponymy set.
7:    end u
8:  end j
9: end i.
```

Now, we get a weighted integrated tree. The integrated tree is shown as fig.3. For nodes *h* and q_1, which are separated by node f, the similarity is $sim_1(q_1, f) \bullet sim_1(f, h)$. If there are many paths between *h* and q_1, then we choose that one which has the largest weight.

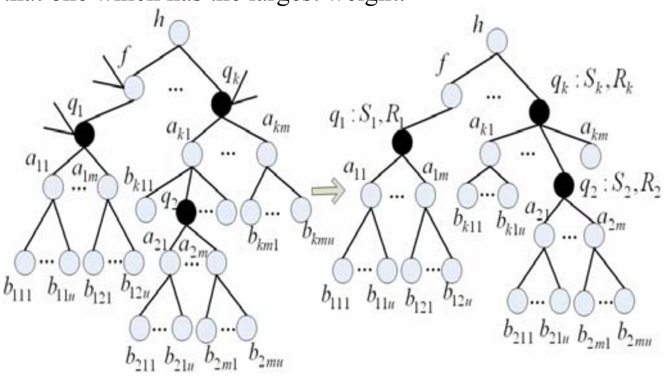

Figure 3. Tree of associational semantics model

C. Supplementation of Associational Semantics

So far, we get semantic relations of words by calculating the similarities between them. Usually, the higher the similarity between two words is, the closer their semantic relationship is, such as 'lady' and 'woman'. However, some words have close associational semantics too, though there is a low similarity between them, such as 'doctor' and 'patient'. These words should be added to query set to reflect semantic relationships between keywords.

In this subsection, we use a type of mutual information with attenuation factor to select those relative words which have close correlation instead of similarity. Let *DMI(x,y)* denote the mutual information. Let *rel(Q,z)* stand for the semantic relation of the query set *Q* and word *z*. In the set $R = <r_1, r_2, \ldots r_n>$, the r_i is such a word which have close correlation with the original keywords and is not in the set *S* or hypernymy/hyponymy set. The *R* is got from returned documents of initial retrieval based on MI. The formulas of calculating *rel(Q,r)* are shown as follows.

$$rel(Q,r) = \sum_{j=1}^{n} DMI(q_j, r) \quad (1)$$

$$DMI(q_j, r) = MI(q_j, r) \cdot D(q_j, r) \quad (2)$$

$$MI(q_j, r) = \ln(\frac{p(q_j, r)}{p(q_j) \cdot p(r)}) \quad (3)$$

$$D(q_j, r) = e^{-\alpha \cdot Dis(q_j, r)} \quad (4)$$

The value of *rel(Q, r)* should be normalized to (0...1). $MI(q_j,r)$ is a ordinary mutual information, and $Dis(q_j,r)$ denotes a minimum value of the number of words between q_j and *r* in a context window. α is an adjustable parameter. In our experiment, α is set to 1. $D(q_j,r)$ is a attenuation factor which represents composite structure of words q_j and *r*. The less the space between q_j and *r* is, the more correlation q_j and *r* have.

We define T_4 as the scheduled threshold for the semantic relation. If $rel(Q, r) \geq T_4$, then add r into *R*, and assign *rel(Q, r)* to *r*, as its weight. Now, we complete the expansion of original query, and get a new set *Q'*, the extended set of keywords. $Q' = Q \cup R \cup S \cup H_j \cup H_d$ Where *Q* is original query, *R* is relative words set, *S* is synonymy set, H_j is q_j's hypernymy set H_j, H_d is q_j's hyponymy set and a_{ij}'s hyponymy set. For a word *q'* in *Q'*, we can assign the corresponding weight *w* to it as follows:

1: if *q'* is a original keyword in *Q*, then *w* = 1;
2: if two original keywords q_a and q_b have common hyponymy *r*, and *q'* locates in a path from r to q_a and q_b, then *w* = 1;
3: if *q'* is in *S*, then $w = sim_3(q_i, s_{ij})$;
4: if *q'* is q_i's direct child node then $w = sim_1(q_i, a_{ij})$;
5: if *q'* is q_i's indirect child node then $w = sim_1(q_i, a_{ij}) \bullet sim_1(a_{ij}, b_{iju})$;
6: if *q'* is in a path from q_i to the root *h* of the tree, then,

$$w = \prod_{i=1}^{v-1} sim(c_i, c_{i+1})$$

; where *v* is the path length from the *q'* to q_i.

D. Using the Expanded Query

In traditional Vector Space Model, *tfidf* is often used to measure semantic relevance between a term and a document. Where *tf* is the frequency of the term occurring in the document and *idf* is the inverted document frequency of the term. The *tfidf* is usually represented as a form and used in the calculation of semantic relevance. The *tfidf* can be added some weights to improve calculation of semantic relevance.

Our query expansion has not only added some new words into the query set, but assigned a corresponding weight to each word in the set also. When we calculate similarities between query and documents based on *tfidf* in the information retrieval, we can use these weights to represent user's query meanings better. In addition, we will add a factor, namely, the distribution density, into the computation formula for *tfidf*. The distribution density shows how the keywords are distributed in documents. In traditional computation formula for *tfidf*, keywords are presumed to be mutual independence, which neglects some semantic information that users are going to express in the query. Actually, there are internal semantic relations between keywords. The relations can be revealed by analyzing the distribution situation of the keywords in documents, paragraphs and sentences. If many keywords present in the same paragraph instead of scattering in entire chapter, this document should be identified as high similarity.

When scanning documents, a specific size of context window can be used. The size can be fixed-length, as did some systems[11], or variable-length. In this paper, we will adopt a variable-length window because it can reveal distribution density of keywords in documents better.

Let *Sim(Q, d_i)* denote the similarity between query *Q* and the document d_i, then,

$$Sim(Q, d_i) = \sum_{j=1}^{p} S_{win}(j) \qquad (5)$$

$$S_{win}(j) = sum(j) \bullet \sum_{v=1}^{k} w_v \bullet tf_v \bullet idf_v \qquad (6)$$

Where *p* is the number of paragraphs in the document d_i, w_v stands for the weight of the vth keyword, $tf_v idf$ is the traditional *tfidf* of the vth keyword. The *sum(j)* is the occurrence number of keywords in the jth paragraph, which reflects the distribution situation of keywords in documents.

Ⅳ. Experiment

In order to validate the ideas presented above, we set up an experimental framework described as follows.

A. Experimental Design

Our experiments are conducted on a set of advertisement texts for online bookstores. Five hundred and thirty ads are randomly gathered from Internet. After removing some strange or imperfect webs(only seller's name, or size less than 10 words), five hundred and two ad texts are selected and assigned to 7 subjects manually by 3 experts. Where two hundred and sixty ads contain text directory. We select 30 terms from the query logs in our school digital library as query terms. Each term is made of 2~4 keywords. The 20 terms are selected as our query training set, and another 10 terms as our test samples. Table I shows keywords of queries in our experiment, and table II shows the classification of advertisements.

TABLE I.
Keywords of Queries

1	computer, network, teaching material
2	computer system, system architecture, system safety
3	database, tutorial, storing process
4	system development, SDK, software engineering, development plan
5	image processing, multimedia making
6	software development, software test
7	programming, tutorial
8	network security, network management
9	project development, software engineering
10	communication protocols, communication security

TABLE II.
Subjects of Advertisements

Subject	The number of ads
computer theory	49
system software	82
database	94
programming	73
network & data communication	97
image processing	34
software engineering	73

In the area of information retrieval, precision/recall is well accepted evaluation method for the performance of the systems. An ideal information retrieval system is trying to raise the values for both of the two objectives. Because recall can be improved easily by the most methods of query expansion although these methods often draw a large amount of irrelevant documents into returned results, in this paper, we will just use precision, more precisely, Mean of Average Precision[MAP] and prec@20, to evaluate our method.

B. Experimental Results

Firstly, we have to determine factor values for T_1, T_2, T_3 and T_4, which decides how much an original query is expanded. Because we have limited the expansion levels of the tree along with hyponymy dimension and the number of child nodes. For simplicity, we need not find proper values for them. On the contrary, we can just define T_1=0.4, and T_2=0.25, based on our past experience.

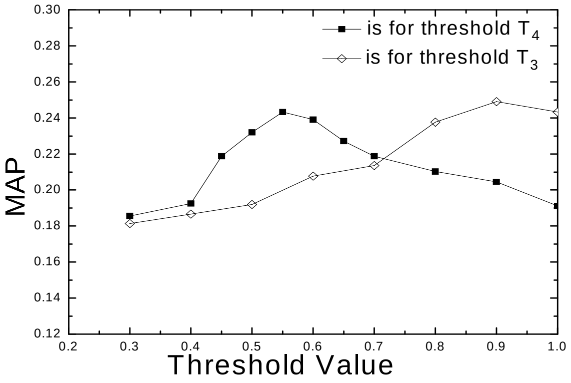

Figure 4. MAP values versus threshold values

Fig.4 shows the curves of MAP values versus factor values for T_3 and T_4. From fig.4, it is clear that MAP has the maximum when T_3=0.9, and that MAP has the maximum when T_4=0.55. The threshold values adopted in the experiment are listed as follows.

TABLE III.
Threshold Values In The Experiment

Threshold	T_1	T_2	T_3	T_4
value	0.4	0.25	0.9	0.55

TABLE IV.
Evaluation Results

Method of query expansion	MAP	Prec@20
tfidf-based method	0.1912	0.452
TASM-based method	0.2191	0.514

The experiments are carried out according to the flow of work shown in fig.1. We compare our experimental results with the ones got from the traditional *tfidf*-based method. Table IV shows their MAP and Prec@20. Experimental results demonstrate about 14.6% precision and 13.7% precision on top 20 documents improvement over the traditional *tfidf*-based method. But it is not a well ahead of other methods. The main reason is that the data set adopted in our experiment is not big enough and some important factors such as the levels of trees, the number of child nodes on the trees have not been optimized well.

Ⅴ. Conclusion And Future Work

In this paper, we propose a method for query expansions. We use WordNet hypernym/hyponymy and synonym relations to expand the query words and use association semantics based on Mutual Information to filter and supplement the expansion. The expansion procedure is constructing a TASM. All query words are put on the TASM and assigned a semantic weight. In the calculation of similarities between queries and documents, the weights and distribution density of words are used to strengthen user's intent.

In our approach, we use Mean Average Precision (AP) as the objective function in choosing the threshold values, and also use MAP and Prec@20 to evaluate the performance of the algorithm. The experiments show that the result of our combined query expansion is better than the traditional method based on *tfidf.*

Some limitations also exist in our current work. The principal problem is that our experiment does not adopt the existing large-scale corpus, such as Text REtrieval Conference[TREC]. We plan to address this problems in our future work. That is, we will carry out the query expansion and documents retrieval on TREC, which enable us to compare our method with more existing approaches.

Acknowledgment

We thank all SQcse project members and our collaborators at the Henan University of Science and Technology, China. This project is supported by the Henan natural science fund committee, award #102102210159.(Research on the semantic query system based on commercial search engines). We particularly thank Wang Pin, who undertook inputing data for our experiments, and Huo Yager, who proofed editing text for this paper.

References

[1] J. Zhang, B, Deng, X. Li. "Concept based query expansion using WordNet", *Proceedings of 2009 International e-Conference on Advanced Science and Technology, AST* 2009, pp. 52-55.

[2] Z. Gong, C. Cheang and L. Hou. "Multi-term web query expansion by WordNet". *Lecture Notes in Computer Science,* v 4080 LNCS, 2006, pp. 379-388.

[3] E. Lesk. "Word-word associations in document retrieval systems". *American Documentation*, 1969, vol.20(1), pp. 8-36.

[4] J. Sparck, O. Barber. "What makes an automatic keyword classification effective". *Journal of the American Society for Information Sciences*, 1971, vol.22(3), pp.166-175.

[5] Y.G. Qiu, H.P. Frei. "Concept-based query expansion". *Proceedings of the 16thACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA,USA*, 1993, pp.160-169.

[6] Y.F. Jing, C. Bruce. "An association thesaurus for Information retrieval". *Proceedings of RIAO 94*, pp.146-160. 1994.

[7] B.M Fonseca, P.B. Golgher, E.S.de Moura, B. Possas, N. Ziviani. "Discovering search engine related query using association rules". *Journal of Web Engineering*, 2004, vol.2(4), pp.215-227.

[8] Aseervatham, Sujeevan. "A concept vector space model for semantic kernels", *International Journal on Artificial Intelligence Tools*, vol.18(2), April 2009, pp.239-272. DOI: 10.1142/S0218213009000123

[9] L. Liu, C.G. Cao, C.X. Zhang, G.G. Tian. "Sense recognition research of hyponymy based on concept space". *Chinese Journal of Computers*, August 2009, pp.1651-1661. DOI: 10.3724/SP.J.1016.2009.01651

[10] E.C.C Van, M.E.M. Van, J.A. Kors, B. Mons, B.J. Van. "Constructing an Associative Concept Space for Literature-Based Discovery". *Journal of the American Society for Information Science and Technology*, vol.55(5), March 2004, pp.436-444. DOI: 10.1002/asi.10392

[11] S. Lu, S. Bai. "Quantitative Analysis of Context Field in Natural Language Processing". *Chinese Journal of Computer*, vol.24, pp. 742-747, July 2001.

Guangjun Huang received his phD degree in Computer Science from Northwestern Polytechnical University, China in 2005. He is a Associate Professor at Henan University of Science and Technology. His current research interests are in area of information retrieval and distributed computing. Dr.Huang became a member of China Computer Society in 2006, a member of IEEE in 2008.

Shuili Wang is a graduate student for a Master's degree at Henan University of Science and Technology. His current research interests focus on information retrieval and software engineering.

Xiaoguo Zhang is a Assistant Professor at Henan University of Science and Technology. His current research interests focus on data mining.

About corresponding author:
Name: Guangjun Huang
Address: School of Electrical and Information Engineering, HeNan University of Science and Technology
48# Xiyuan road, Luoyang, Henan Province, 471003, China
Tel: 86+379+13643893139
Email:guangjunhuang@126.com