Multi-term Web Query Expansion Using WordNet

Zhiguo Gong, Chan Wa Cheang, and Leong Hou U

Faculty of Science and Technology
University of Macau
Macao, PRC
{zggong, ma36600, ma36575}@umac.mo

Abstract. In this paper, we propose a method for multi-term query expansions based on WordNet. In our approach, *Hypernym/Hyponymy* and *Synonym* relations in WordNet is used as the basic expansion rules. Then we use WordNet Lexical Chains and WordNet semantic similarity to assign terms in the same query into different groups with respect to their semantic similarities. For each group, we expand the highest terms in the WordNet hierarchies with *Hypernym* and *Synonym*, the lowest terms with Hyponym and Synonym, and all other terms with only Synonym. Furthermore, we use collection related term semantic network to remove the low-frequency and unusual words in the expansions. And our experiment reveals that our solution for query expansion can improve the query performance dramatically.

1 Introduction

One challenging issue, among others, in information retrieval is the problem caused by word mismatch. That is, the query words may not rightly be contained in the document even though their semantics are highly relevant to the user's need. Evidently, if the word mismatch problem is not appropriately addressed by information retrieval systems, it could degrade their retrieval performance greatly. To deal with this problem, query expansion is one of the promising approaches. Typical methods include *Lexical-Based* [13, 14, 15], *Statistical-Based* [16,17,18], *Query-Log-Based* [19], and *Web Link-Based* [20].

Lexical-Based method utilizes some manually created lexical thesaurus for the expansion. For any term in the query, a list of semantic relevant terms is selected from the thesaurus and then used for the expansion. In such method, the thesaurus used is often collection independent, thus may not catch the dynamic change of the vocabulary used in the collection. Therefore, the effectiveness of such method is often not as expected in practice.

Statistical-Based solutions, on the other hand, describe word relations using their co-occurrences in the collection. Actually, term co-occurrences can be globally extracted in the scope of whole collection (Global Expansion) or locally obtained from the results of initial query (Local Expansion). With these methods, a term is selected for expansion if it has higher degree of co-occurrences with the query terms. The effectiveness of such kind of methods is dependant on the collection. If the size of collection is not huge enough, it may not well capture the relations between terms.

S. Bressan, J. Küng, and R. Wagner (Eds.): DEXA 2006, LNCS 4080, pp. 379 – 388, 2006.

Another problem lies in the fact that term relations captured are only pair based, without a semantic architecture among all the expanded terms. Therefore, it is hard to control the mutual impairing caused by multiple terms in the query.

Query-Log-Based expansion methodologies describe term-term relations by introducing users' click-though activities. In other words, term t_2 can be used to expand query term t_1 if, historically, many users, who query term t_1, have clicked documents which contain t_2 in the results. In general, users' click-through activities are captured in the system logs. As a matter of the fact, users click-though information can only be considered as implicit indicators for the term relations. That is, a user may click a result document just because he is motivated by any other reasons than relevant. As the result, it may provide poor performance if less people previously query the words. This is common especially when the system is just created.

Web Link-Based solutions expand Web queries with a thesaurus which is constructed by using links of the Web. To create the thesaurus, Web pages as the training set are selected manually. And the semantic of a target Web page of a link is represented as the words or concepts appearing in the anchor texts in the source page of the link. Then the semantic relations among the words or concepts are derived using the links.

The method in this paper belongs to the first type. However, we have two important improvements in the expansion: (1) clustering all terms of a query into different groups by their semantic similarities, then expanding each group by taking into account their positions in WordNet [6]; (2) reducing noise terms in the expansion by term co-occurrences supported by the collection.

In our approach, *Hypernym/Hyponymy* and *Synonym* relations in WordNet is used as the basic expansion rules. Then we use WordNet Lexical Chains and WordNet semantic similarity to assign terms in the same query into different groups with respect to their semantic similarities. For each group, we expand the highest terms in the WordNet hierarchies with *Hypernym* and *Synonym*, the lowest terms with Hyponym and Synonym, and all other terms with only Synonym. In this way, contradictory caused by full expansion can be well controlled. Furthermore, we use collection related term semantic network to remove the low-frequency and unusual words in the expansions. And our experiment reveals that our solution for query expansion can improve the query performance dramatically.

In reminder of this paper, section 2 provides our detail methodologies for query expansion with WordNet::Similarity and reduction using *TSN*. The experiment results are illustrated and discussed in section 3. Finally, we conclude our work in section 4.

2 Expansion Method

In this section, after a brief introduction of our previous work for single word query expansions, we address our multi-term query expansion methodologies in detail.

2.1 Single Term Query Expansion

In [7], we have addressed our solutions for single word query expansions using WordNet and TSN (Term Semantic Network). WordNet organizes words or concepts lexically into hierarchies. Figure 1 is a typical example in which term ‘Software’ can be semantically expanded along three chains, say, Hypernym (i.e. ‘code’), Hyponym (i.e. ‘program’, ‘freeware’, ‘shareware’, ‘upgrade’, ‘groupware’) and Synonym (i.e. ‘software system’, ‘software package’, ‘package’). Actually, Hypernym is the abstractive concepts of the term while Hyponym, reversely, includes specific concepts of the terms. And Synonym contains all the synonyms of the term. However, their impacts for the expansion are different in letter of retrieval performance.

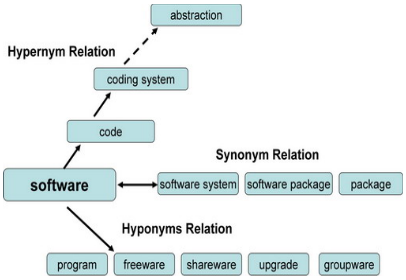


Fig. 1. An Example for WordNet

According to our experiments in previous research, WordNet may bring many noises for the expansion because of its collection independent characteristic. And it may not catch current state of words and their relationships since the explosive increase of the Web. To overcome those problems, collection-related *TSN* (Term Semantic Network) is created with respect to word co-occurrence in the collection. We use *TSN* both as a filter and a supplement for WordNet.

For any term *t*, let $Hyper_t$, $Hypo_t$, Syn_t and TSN_t stand for the concept sets of its Hypernym, Hyponym, Synonym, and Top-k of TSN respectively. Let $R(p|q)$ be the rank of Web page *p* with respect to query *q*. Ranking model *tf* or *tf*idf* is popularly employed in the information retrieval world because of its robustness and simplicity [8]. And in our Web image search system, we modify model *tf* into model *ttf* by incorporating term *t*’s locations in *p* with respect to the corresponding Web image [5, 11]. For any single word query *t*, we define its expanded rank function $ER(p|t)$ as

$$ER(p\,|\,t) = R(p\,|\,t) + \alpha \cdot \sum\nolimits_{z \in Hyper_t} R(p\,|\,z) + \beta \cdot \sum\nolimits_{z \in Hypo_t} R(p\,|\,z) + \gamma \cdot \sum\nolimits_{z \in Syn_t} R(p\,|\,z) + \delta \cdot \sum\nolimits_{z \in TSN_t} R(p\,|\,z) \quad (1)$$

where α, β, γ and δ are factors used to indicate different effects from different expansion directions. In our work, we suppose the expansion along each dimension is independent. And we use Average Precision (*AP*) of the retrieval as the objective function in determining the optimal values of those factors which can maximize *AP* value. Table 1 shows the optimal factor values with their corresponding *AP* values in our Web image retrieval system [7].

Table 1. Factor Values and Average Precision

Factor	*Values*	*Average Precision*
α (Hypernyms)	0.47	0.2406
β (Hyponyms)	0.84	0.3888
γ (Synonyms)	0.70	0.3404
δ (Top-k of TSN)	0.94	0.3559

In Web image retrieval system [7], we combined the query expansions along each semantic dimension as our overall solution. Our experiments reveal that the combined expansion can provide a satisfied result for the Web query performance. However, previous method ignored the mutual affections among the terms in the same query.

2.2 Multi-term Query Expansion

Even though most of Web users search the Web with only one word, we still find many queries with multiple terms. For example, a user uses a pair words (computer, speaker) as one query to search Web images. Our previous expansion method will automatically expand these two words with three semantic relations independently. As a result, the expanded query will contain too many words which may include many noise words, thus, reduce the precision of the query results. For example, a Web user may use *q*=(software, groupware) as the query. With single word expansions, query *q* will be expanded to include all the words or concepts from both 'software' and 'groupware's WordNet expansions. However, as in figure 1, 'groupware' is in the Hyponym of 'software'. The user, who uses 'groupware' to combine 'software', implicitly wants to exclude other words in the Hyponym of 'software' for the query. Therefore, the overall expansions of these two words may bring many words which contradict to the user's query intention.

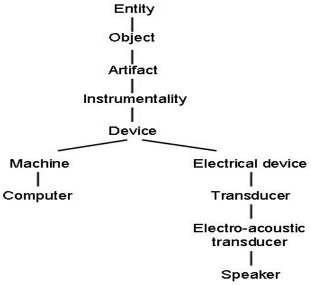


Fig. 2. Group Terms in the WordNet

Figure 2 shows another situation, in which "computer" and "speaker" have the closest super-ordinate class "device" along the WordNet chains. Even though 'speaker' is not in the direct Hyponym of 'computer', it is sill located in the lower level with respect to 'computer' in the WordNet hierarchies. Therefore, we also suppose 'speaker' is a restraint for 'computer' in Hyponym expansions of 'computer'. Reversely, 'computer' is taken as the constraint of 'speaker's Hypernym expansions.

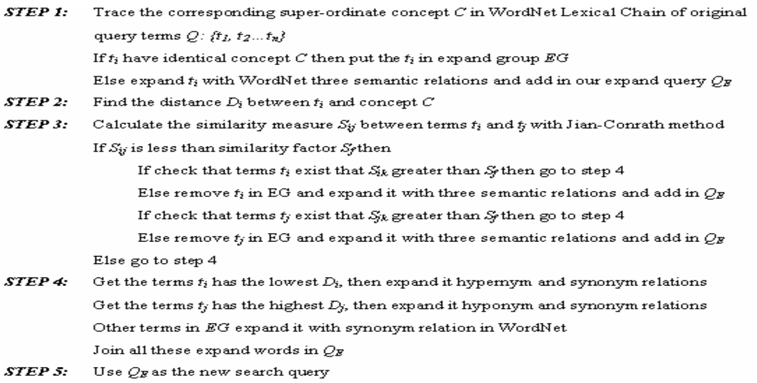
In this work, we use Jian-Conrath [4] to measure the distances of two words in WordNet. In fact, this measure method combines WordNet lexical taxonomy structure with corpus statistical information such that the semantic distances between nodes in the semantic space constructed by the taxonomy can be better quantified with the computational evidence derived from a distributional analysis of corpus data.

Jian-Conrath approach uses the notion of information content, but in the form of the conditional probability of encountering an instance of a child-synset given an instance of a parent-synset. Thus the information content of the two nodes, as well as that of their most specific subsumer, is taken into account in the measure calculations. Notice that this formula measures semantic distance in the inverse of similarity as:

$$Dist(c_1, c_2) = 2\log(p(lso(c_1, c_2))) - (\log(p(c_1)) + \log(p(c_2)) \qquad (2)$$

where c_1 and c_2 are synsets, *p(c)* is the probability of encountering an instance of a synset *c* in some specific corpus, *lso*(c_1, c_2) is the similarity between two concepts lexicalized in WordNet to be the information content of their lowest super-ordinate (most specific common subsumer). Therefore, we could use this approach to measure the semantic strength between tow words.

The larger the *Dist*(t_1, t_2) is, the farer term t_1 to term t_2 is in the WordNet hierarchies. In case $t_1 > t_2$ (t_1 is located in a higher level than t_2 in WordNet), we do not expand Hyponym for t_1 and Hypernym for t_2 as the reason we discussed previously. Going back for our last example, we only expand "computer" with Hypernym and Synonym relations, and "speaker" with Hyponym and Synonym relations. In fact, we will assign terms in the same query *q* into groups. Within each group, terms are clustered with respect to *Dist*(*t1*,*t2*), and we expand Synonym for all terms and Hypernym only for the words on the highest level in the WordNet hierarchies, and Hyponym only for the words on the lowest level of WordNet hierarchies. Figure 3 provide our detail expansion algorithm.



```
STEP 1:  Trace the corresponding super-ordinate concept $C$ in WordNet Lexical Chain of original
         query terms $Q: \{t_1, t_2 \dots t_n\}$
         If $t_i$ have identical concept $C$ then put the $t_i$ in expand group $EG$
         Else expand $t_i$ with WordNet three semantic relations and add in our expand query $Q_E$
STEP 2:  Find the distance $D_i$ between $t_i$ and concept $C$
STEP 3:  Calculate the similarity measure $S_{ij}$ between terms $t_i$ and $t_j$ with Jian-Conrath method
         If $S_{ij}$ is less than similarity factor $S_f$ then
             If check that terms $t_i$ exist that $S_{ik}$ greater than $S_f$ then go to step 4
             Else remove $t_i$ in EG and expand it with three semantic relations and add in $Q_E$
             If check that terms $t_j$ exist that $S_{jk}$ greater than $S_f$ then go to step 4
             Else remove $t_j$ in EG and expand it with three semantic relations and add in $Q_E$
         Else go to step 4
STEP 4:  Get the terms $t_i$ has the lowest $D_i$, then expand it hypernym and synonym relations
         Get the terms $t_j$ has the highest $D_j$, then expand it hyponym and synonym relations
         Other terms in $EG$ expand it with synonym relation in WordNet
         Join all these expand words in $Q_E$
STEP 5:  Use $Q_E$ as the new search query
```

Fig. 3. Algorithm of our expand method

Below we use some examples as illustrations for the algorithm. Let *q*=('Macau', 'camera', 'photo') be a three-term query. Our expansion algorithm divides them into two groups by checking their similarities. One group contains the word "Macau", another one contains "camera" and "photo". According to the similarity measure, "Macau" does not have similarity relation with other two words. Therefore, in this example, we expand 'Macau' via three WordNet chains. In the second group, "camera" and "photo" have a high degree in similarity and they are under the same concept in WordNet hierarchies, with 'camera' > 'photo'. So the system expands "camera" through hypernym and synonym relations and "photo" through hyponym and synonym relations (Figure 4).

Figure 5 provides another example, where query q=('movie', 'camera', 'character'). By tracing them in WordNet, these three words are under the same concept and with close similarity values mutually. Thus, our algorithm keeps them in one group, with WordNet hierarchal levels like 'movie'>'camera'> 'character'. In this case, the algorithm only expands "movie" with hypernym and synonym relations because of its highest WordNet hierarchical level within the group, and "character" is expanded in hyponym and synonym relations due to its lowest level within the group. The word "camera" is only expanded in synonym relation because its location in WordNet is between "movie" and "character".

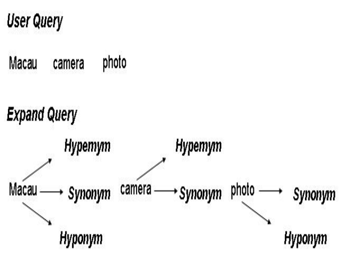


Fig. 4. Query expansion with two relative terms

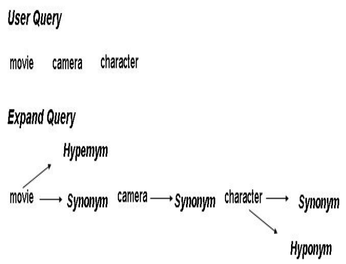


Fig. 5. Query expansion with three relative terms

2.3 Similarity Threshold for Grouping Words

As in our discussions of last section, term grouping in a multi-term query is a critical step in our expansion algorithm. In this section, we are going to determine the optimal similarity threshold for grouping terms in the query.

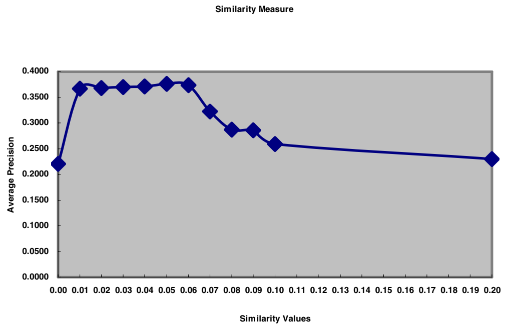


Fig. 6. Average precision versus Similarity Values

To determine the threshold value for term grouping, we select *AP* (average precision) as the objective function [8,9]. In other words, the optimal value for the similarity threshold should maximize *AP* values of retrievals. In this work, we use 40 single-word queries, 40 two-word queries, 20 three-word queries, 10 four-word queries and 10 five-words as our sample queries. And we select about 60% of them as our training set for the threshold determination. In the same query, if the similarity of two terms is over the threshold (*UD*) we assign them in the same group. Figure 6 is the performance of average precision via similarity threshold (*UD*). From this figures, it is obvious that the retrieval performance reaches its maximum when threshold *UD* is at 0.05. As a matter of the fact, in our sample data there are no pair of words whose similarity is greater than 0.2, so we ignore the figure plot when similarity range are between 0.2 to 1. Furthermore, we could find the retrieval performance drops down dramatically when the value is over 0.05.

2.4 Query Reduction

In general, query expansion using a thesaurus may be expanded to include too many words. And some of them are low-frequency and unusual words in the collection. Those unusual words may bring in some noise and decrease retrieval performances. Therefore, it is important to exclude noise words during query expanding. In our system, a term semantic network (*TSN*) is extracted from the collection. Actually, *TSN* is a direct graph with words as its nodes and the associations as the edges between two words.

To extract TSN from the collection, we use a popular association mining algorithm – Apriori [12] — to mine out the association rules between words. Here, we only consider one-to-one term relationship. Two functions—*confidence* and *support*— are used in describing word relations. We define *confidence* (*conf*) and *support* (*sup*) of term association $t_i \rightarrow t_j$ as follows, let

$$D(t_i, t_j) = D(t_i) \cap D(t_j) \tag{3}$$

where $D(t_i)$ and $D(t_j)$ stand for the documents including term t_i and t_i respectively. Therefore, $D(t_i) \cap D(t_j)$ is the set of documents that include both t_i and t_j. We define

$$Conf_{ti->tj} = \frac{\| D(t_i, t_j) \|}{\| D(t_i) \|} \tag{4}$$

where $\| D(t_i, t_j) \|$ stands for the total number of documents that include both term t_i, and t_j; and $\| D(t_i) \|$ stands for the total number of documents that include t_i ,

$$Sup_{ti->tj} = \frac{\| D(t_i, t_j) \|}{\|D\|} \tag{5}$$

where $\|D\|$ stands for the number of document in the database.

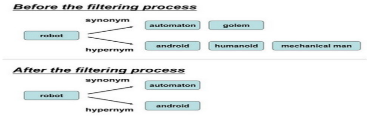


Fig. 7. Keyword filtering process of word "robot"

In this paper, we only remain the expanded words which have minimum confidence over 0.1 and support over 0.01 with the original query keyword into our query expansion. As the keyword "robot" in Fig 7, we filter out the words "golem, humanoid, mechanical man".

3 Evaluation

The crawler of our system gathered about 150,000 Web pages with a given set of seeds which are randomly selected from dot-com, dot-edu and dot-gov domains. After the noise images (icons, banners, logos, and any image with size less than 5k) removed by the image extractor, about 12,000 web images embedded in the Web pages are left. In order to calculate the precision/recall value, our system needs domain experts to annotate the sample Web images with their semantics. Our system provides a user-friendly interface, to let the experts define the corresponded meanings for each image easily by using the mouse [5]. Then the human experts are assigned to define the subjects of the Web images manually. And sometimes, more than one subject is defined for the same images. For example, concepts 'Laptop', 'Notebook' and 'Computer' may be annotated to the same Web image.

As in Figure 9, we have used different expansion method in our experiment in order to compare their performances. In the evaluation, we use the remaining 40% of the sample queries as the testing queries. Below are the descriptions for different expansion models:

No Expand – use original queries in the testing, without any expansion.
All Expand – use WordNet three semantic relations (Hypernym, Hyponyms, Synonym) to expand original queries fully, without word grouping.
UD_Expand – we treat all terms in the same query as one group without concerning their similarities, and only expand the highest level terms with hypernym and synonym relations and the lowest level terms with hyponym and synonym relations. Other terms only expand its synonym relation.
UD~0.05 – Group terms with the similarity threshold as 0.05 in the same query. In each group, we expand the highest terms with hypernym and synonym and lowest terms with hyponym and synonym, all others with only synonyms.
Reduction – This model is UD*~0.05 + Term Reduction*. We use *UD~0.05* for term expansion, and remove noise words using *TSN.*

As revealed in Figure 8, even though *ALL_EXPAND* can improve recalls of queries a little bit, however, its retrieval precision is the lowest among all the models. The reason is due to the fact that, besides the noise words, there are too many words included in the expansion, and some of them are contradictory with each other. *UD_EXPAND* model has improved both precision and recall comparing with *NO_EXPAND* model. It impose some constraint on the expansion scope, thus reduce some contradictories among words in contrast to *ALL_EXPAND* method. *UD~0.05* model produce a quite good performance comparing with both *ALL_EXPAND* and *UD_EXPAND* models. That means grouping terms with similarity threshold UD=0.05 is both critical and necessary in improving the retrieval performances. This model can effectively reduce the contradictories among words when expanding the queries. This model overcomes the weaknesses of two extremes *–ALL_EXPAND* and *UD_EXPAND*. Finally, *REDUCTION* model is the best in term of retrieval performances among all those models. As we know, it enhances *UD~0.05* by further removing words which have lower associations with the words in the original query which may disturb the searching.

Through our discussions above, we can conclude that query expansions with WordNet yield significant increases in the number of correct documents retrieved and in the number of answerable queries, and query expansions followed by reduction makes even more substantial improvements.

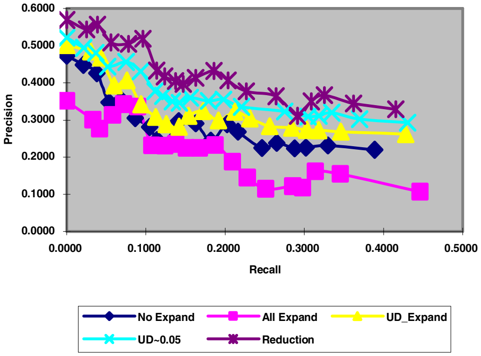


Fig. 8. Performance of Multi-term Query Expansion method

4 Conclusions

In this paper, we propose a method for multi-term query expansions. We use WordNet noun hypernym/hyponymy and synonym relations between words as the base expansion dimensions. In our approach, we divide terms in the same query into groups with respect to semantic similarities between terms. Within each group, the terms are closely related in semantics. We determine expansion dimensions for each word in the same group by their relative positions in the WordNet hierarchies. We only expand the top words with Hypernym and Synonym, the bottom words with Hyponym and Synonym, all other words with only Synonyms. By this way, the contradictories among words in the expansions can be well controlled, thus retrieval performances can be improved. Furthermore, in order to avoid noise words in the expansions, we apply term co-occurrence information further to remove unusual words during query expansion processing.

References

1. Ted Pedersem, Siddharth Patwardhan and Jason Michelizzi, *WordNet::Similarity – Measuring the Relatedness of Concept*, In Proc. of Fifth Annual Meeting of the North American Chapter of the ACL (NACCL-04), Boston, MA, 2004.
2. WordNet::Similarity, http://search.cpan.org/dist/WordNet-Similarity/
3. Alexander Budanitsky and Graeme Hirst, *Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures*, In NAACL Workshop on WordNet and Other Lexical Resources, 2001.
4. Jay J. Jiang and David W. Conrath, *Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy*, In the Proceedings of ROCLING X, Taiwan, 1997

5. Zhiguo Gong, Chan Wa Cheang, Leong Hou U, *Web Query Expansion by WordNet*, DEXA 2005
6. Miller, G. A., Beckwith, R., Felbaum, C., Gross, D., and Miller, K., *Introduction to WordNet: An On-line Lexicala Database,* Revised Version 1993.
7. Zhiguo Gong, Leong Hou U and Chan Wa Cheang, *An Implementation of Web Image Search Engine*, Digital Libraries: International Collaboration and Cross-Fertilization: 7th International Conference on Asian Digital Libraries, ICADL 2004, Shanghai, China, December 13-17, 2004. Proceedings Pages:355 – 367
8. R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley, 1999
9. R. K. Sriari, Z. Zhang and A. Rao, *Intelligent indexing and semantic retrieval of multimodal documents*, Information Retrieval 2(2), Kluwer Academic Publishers, 2000, pp. 1-37.
10. Hang Cui, Ji-Rong Wen, Jian-Yun Nie, Wei-Ying Ma, *Query Expansion by Mining User Logs*, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 4, July/August 2003, pp. 829-839
11. Zhiguo Gong, Leong Hou U and Chan Wa Cheang, Text-Based Semantic Extractions of Web Images, To appear in Knowledge and Information Systems: An International Journal, Springer.
12. R. Agrawaland R. Srikant, “Fast Algorithms for Mining Association Rules,” *Proc. 20th Int’l Conf. Very Large Data Bases, (VLDB)*, Sept. 1994.
13. E. M. Voorhees, Query Expansion using Lexical-Semantic Relations. In Proceedings of the 17th ACM-SIGIR Conference, pp. 61-69, 1994.
14. A.F. Smeaton and C. Berrut. Thresholding postings lists, query expansion by word-worddistance and POS tagging of Spanish text. In Proceedings of the 4th Text Retrieval Conference, 1996.
15. Oh-Woog Kwon, Myoung-Cheol Kim, Key-Sun Choi. Query Expansion Using Domain-Adapted Thesaurus in an Extended Boolean Model. In Proceedings of ACM CIKM’94. pp. 140-146.
16. Yonggang Qiu and H.P. Frei. Concept Based Query Expansion. In Proceedings of ACM-SIGIR’93. pp. 160-169.
17. Jinxi Xu and W. B Croft. Improving the Effectiveness of Information Retrieval with Local Context Analysis. ACM Transactions on Information Systems, Vol. 18, No. 1, January 2000, pp. 79-112.
18. Jing Bai, Dawei Song, Peter Bruza, Jian-yun Nie, and Guihong Cao. Query Expansion Using Term Relationships in Language Models for Information Retrieval. In Proceedings of ACM CIKM’05, pp. 688-695.
19. B. Billerbeck, F. Scholer, H.E. Williams, and J. Zobel. Query Expansion Using Associated Queries. In Proceedings of ACM CIKM’03, pp. 2-9.
20. Z. Chen, S. Liu, W. Liu, A. Pu, and W. Ma. Building a Web Thesaurus from Web Link Structure. In Proceedings of ACM SIGIR’03, pp. 48-55.