Multi-strategy Query Expansion Method Based on Semantics

Li Li[1, 2], Hongbing Wang[1, 3]
[1]School of Computer and Communication Engineering
University of Science and Technology, Beijing
[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science
Beijing, China
[3] Department of Industrial and Systems Engineering
North Carolina State University, Raleigh, USA
86793343@qq.com, liliustb@126.com

Journal of Digital Information Management

ABSTRACT: *Query expansion adds related words to a user query in order to improve retrieval results. It's an important step in information retrieval. Most of current query expansion methods pay attention to specific expansion strategies or algorithms, while neglecting the query itself. In reaction to the phenomenon, a multi-strategy query expansion method based on semantics was proposed. This method started by analyzing the semantic structure of user query, and adopted corresponding strategy to select expansion terms. The expansion words are derived from three parts: WordNet, massive web page set and search engine performance evaluation data, which were merged semantically in each expansion algorithm later. The experiment showed this method can improve retrieval results to some extent.*

Subject Categories and Descriptors
H.3.3 [Information Search and Retrieval]; Query Formulation:
I.2.7 [Natural Language Processing]

General Terms: Information Retrieval, Query Expansion

Keywords: Semantic Analysis, Semantic Language, WordNet, Related Words

Received: 6 September 2013, Revised 12 January 2014, Accepted 21 January 2014

1. Introduction

There are differences between users' queries submitted to search engine and their true intentions, which results in bad retrieval result. In query expansion, related words are added to user's original query and form a longer and more precise query to express users' retrieval intentions.

The key to query expansion is how to find words related to original query. Different approach has different strategy for finding related words. Query expansion methods based on corpus extract related words from document set, for example, global analysis based on the all documents [1] and local analysis based on partial documents [2, 3, 4]; Query expansion methods based on semantics find related words from linguistic resources such as WordNet, HowNet and domain ontology [5]; Query expansion methods based on user feedback obtain related words from query log or browser history [6]; or the improvement and combination of the above methods [7, 8]. These methods achieve better retrieval result to a great extent, but most of them concentrate on the concrete strategy or algorithm for related words extraction from corpus or linguistic resources, and often ignore the query itself. Most of these methods do not analyze the feature of user query deeply.

This paper proposed a multi-strategy query expansion method based on semantics. This method started by analyzing the semantic structure and type of user query, and adopted corresponding strategy to select expansion words. The expansion words are derived from three parts: noun and verb expansion words base on WordNet, related words based on massive web page set and related words

based on search engine performance evaluation data. These three parts of expansion terms were merged semantically in each expansion algorithm later.

2. Analysis of query feature in search engine

By analyzing real example query log(SogouQ2008) distributed by Sogou Labs, the authors summarize several features of original query.

(1) Queries are often expressed in natural languages.

(2) In form, queries are often several nouns or noun phrases (such as "*Chinese novel WULINMENGZHU* "), simple verb phrases (such as "*loot relief supplies*", "*ban Sharon Stone*"), or simple sentences (such as "*How to match summer clothes*" and "*what time will Taiwan return to China*").

(3) In content, queries are often person name(such as "*Fangfei LIU*"), place name(such as "*Wuhu FANGTE*"), organization name (such as "*Zhuhai Health Center*"), product or its attribute (such as "*LaCrosse price*"), or events (such as "2008 *earthquake rescue show*").

(4) Queries are usually short. For example, the above example query log file contains 10,000 queries. Each query has 6.84 characters or 3.56 words in average. There are 16135 nouns, 6271 verbs among all the words.

Based on the above features, this paper lays emphasis on verb, noun and simple sentences in this paper.

3. Multi-Strategy Query Expansion Method Based on Semantics

Several facts are considered as follows when expanding user query:

(1) The semantic structure of user query must be analyzed and understood before expansion. Semantic language [9] is adopted to analyze and express user query.

(2) Most content of user queries are nouns, verbs and simple sentences. The kernel of a sentence is verb [10], too. So this paper focuses on the expansion of noun and verb, and constructs the expansion word set of noun and verb based on WordNet.

(3) In addition to synonymy, antonym, hypernym, hyponym, meronym, holonym of nouns, and synonymy, antonym, hypernym, troponym, entailment of verbs, correlation between words is a useful reference in query expansion. This paper finds semantic correlation between words from a large-scale web set, thus constructing related word set based on massive web page set.

(4) Users' evaluation of search results page fully shows the correlation between user query and the search results page, so related words can be achieved from the search engine performance evaluation data, thus constructing related word set based on search engine performance evaluation data.

Based on the above facts, the overall process of multi-strategy query expansion method based on semantics is shown in Figure 1.

3.1 Semantic analysis of user query

To be expanded, user query should be analyzed and represented as accurately as possible. The analysis and representation of user query is based on semantic language [9]. The main idea of semantic language is as follows: The sense of a sentence is called SS. An element to express a meaning in an SS is called semantic element (SE). The representation of an SE in a natural language-I, such as English, Chinese…, is called the representation of SE in Language-I (SER_i). Different languages can be translated into each other because all SSs can be represented in different languages. A high speed semantic analysis method was proposed based on semantic language [11].

The SEs of a user query are extracted using the above semantic analysis method on the basis of SER base built in advance, thus forming the semantic structure expression of user query. After verifying completeness and deleting repeated or redundant SEs, the basic SER base of user query is constructed [12].

Two kinds of user queries should be preprocessed:

(1) Queries which are composed of over one words should be split into multiple sub-queries. Each sub-query is processed separately. For example, "*cause of Wenchuan earthquake The Three Gorges Dam*" are split into two sub-queries: "*cause of Wenchuan earthquake*" and "*The Three Gorges Dam*".

(2) Queries with "*noun + verb*" type are rewritten as "*verb + noun*" to ensure their senses. For example, "*Xingmengyuan watch online*" is rewritten as "*watch Xingmengyuan online*", and "*MD download*" is rewritten as "*download MD*".

To analyze the characteristics of user queries, 146 user queries are randomly chosen from real example query log (SogouQ2008). There are 121 queries after eliminating repeated or illegal queries. This paper pays more attention to SEs of type N, VP and J, which account for 78.60%, 9.76% and 4.19% respectively. SEs of these three types totally account for 92.55% of all SEs.

SE of type J means user submits a query of full sentence, for example, "*how to do when meeting the earthquake in a tower*", "*Where are volcanos located in the world*?", "*What are earthquake precursors*?" and "*Yao Ming beat Kobe*".

According to different tones, sentences of type J are divided into declarative sentence, interrogative sentence, exclamatory sentence and imperative sentence [10]. Sentences of type J, which user submits to search engine, are usually declarative sentences and interrogative sentences.

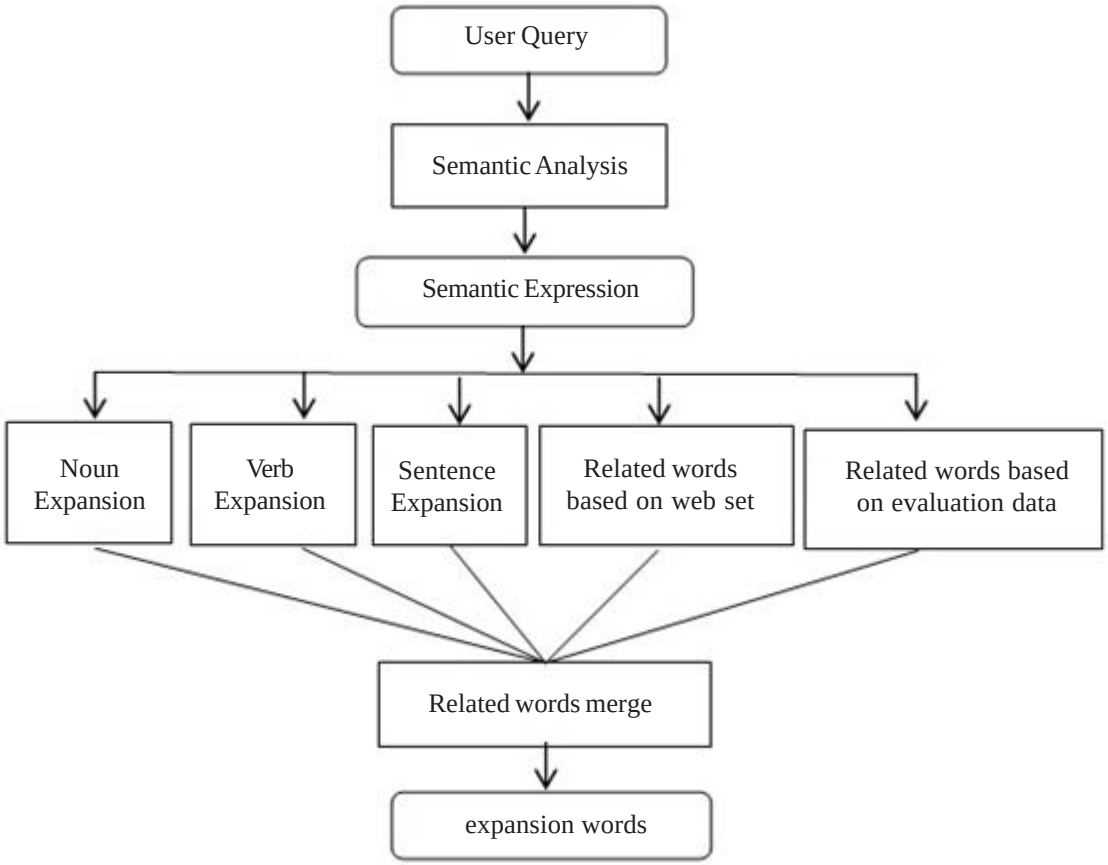

Figure 1. Overall Process of Multi-strategy Query Expansion

The semantic structure of a declarative sentence can be acquired with the high speed semantic analysis method based on semantic language [11]. For example, the semantic structure expression of "*Yao Ming beat Kobe*" is "J: (B_{eat} ($Y_{ao\ Ming}$, K_{obe}))".

Interrogative sentence should be analyzed first. It's important to know clearly which part is questioned: the whole sentence, core verb, the first parameter of core verb, the second parameter of core verb, qualifier of noun, quantity, quantifier, time, location, condition, cause, instrument, method, etc. For example, the questioned part of "*how to do when meeting the earthquake in a tower*" is method, the questioned part of "*Where are volcanos located in the world*?" is location, and the questioned part of "*What are earthquake precursors*?" is the second parameter of core verb "*is*".

The semantic structure of an interrogative sentence should be rewritten according to its questioned part. For example, the semantic structure expression of "*how to do when meeting the earthquake in a tower*" is rewritten as "J: ($H_{ow\ to\ do}$ T: (W_{hen} (J: (M_{eet} (T_{ower}, $E_{arthquake}$)))))", the semantic structure expression of "*Where are volcanos located in the world*?" is rewritten as "N: (o_f (V_{olcano}, $L_{ocation}$)", and the semantic structure expression of "*What are earthquake precursors*?" is rewritten as "N: (o_f ($E_{arthquake}$, $P_{recursor}$)". A query of interrogative sentence is usually rewritten as a noun phrase.

3.2 Noun expansion based on WordNet

Noun and verb are expanded on the basis of WordNet (version 3.1) in this paper.

3.2.1 Translation between English words and Chinese words

WordNet is built in English, but this paper mainly studies Chinese query expansion, so translation between English words and Chinese words is necessary. Microsoft translator (microsoft-translator-java-api-0.6.2) is adopted here, through which any Chinese word can be translated into English word, and vice versa.

3.2.2 Form noun expansion word set

There are 82192 noun synset and 117953 nouns. The synset of a noun can be obtained in noun index file (index.noun). The hypernym, hyponym, meronym and holonym of noun synset can be obtained in noun data file (data.noun).

Supposing there is a noun word W_i, its sense set, SS_i = $\{s_{i1}, s_{i2}, \ldots, s_{iNi}\}$, can be obtained in noun index file (index.noun). Here S_{ij} is the jth sense of W_i, and it is expressed in the sense_number in WordNet.

ESS_i, the expansion sense set of W_i, should include the hypernym, hyponym, meronym and holonym of each S_{ij}. The direct hypernym and hyponym of W_i are selected here

to avoid excessive expansion. Suppose the $hyponSS_{ij}$ is the hypernym sense set of S_{ij}, $hyponSS_{ij}$ is the hyponym sense set of S_{ij}, $meronSS_{ij}$ is the meronym sense set of S_{ij}, and $holonSS_{ij}$, is the holonym sense set of S_{ij}. The expansion sense set of S_{ij}, $ESS_{ij} = hyperSS_{ij} \cup hyponSS_{ij} \cup meronSS_{ij} \cup holonSS_{ij}$. The expansion sense set of W_i, $ESS_i = SS_i \cup hyperSS_{ij} \cup hyponSS_{ij} \cup meronSS_{ij} \cup holonSS_{ij}$, in which $hyperSS_i = hyperSS_{i1} \cup hyperSS_{i2} \cup \ldots \cup hyperSS_{iNi} = \{h1S_{i1}, h1S_{i2}, \ldots, h1S_{iM1}\}$, $hyponSS_i = hyponSS_{i1} \cup hyponSS_{i2} \cup \ldots \cup hyponSS_{iNi}$, $meronSS_i = meronSS_{i1} \cup meronSS_{i2} \cup \ldots \cup meronSS_{iNi}$, $holonSS_i = holonSS_{i1} \cup holonSS_{i2} \cup \ldots \cup holonSS_{iNi}$.

All the words of each sense $h1S_{ij}$ can be obtained in noun data file (data.noun) and they can be translated into Chinese, thus forming $h1W_{ij}$, the Chinese word set of $h1S_{ij}$. So the Chinese word set of $hyperSS_{ij}$, $hyper(w_i) = h1W_{i1} \cup h1W_{i2} \cup \ldots \cup h1W_{i3}$. The same procedure may be easily adapted to obtain $sysW_i$, $hyponW_i$, $meronW_i$ and $holonW_i$.

So the noun expansion word set of W_i, $EWS(w_i) = \{sys(w_i), hyper(w_i), hypon(w_i), meron(w_i), holon(w_i)\}$.

3.3 Verb expansion based on WordNet

There are 13789 verb synset and 11540 verbs. The synset of a verb can be obtained in verb index file (index.verb). The antonym, hypernym, troponym, entailment of verb synset can be obtained in verb data file (data.verb).

Supposing there is a verb word W_i, its sense set, $SS_i = \{s_{i1}, s_{i2}, \ldots s_{iNi}\}$, can be obtained in verb index.verb. Here S_{ij} is the j^{th} sense of W_i, and it is expressed in the sense_number in WordNet.

ESS_i, the expansion sense set of W_i, should include the hypernym, troponym and entailment of each S_{ij}. The direct hypernym and troponym of W_i are selected here to avoid excessive expansion. Suppose the $hyperSS_{ij}$ is the hypernym sense set of S_{ij}, $troponSS_{ij}$ is the troponym sense set of S_{ij} and $entailSS_{ij}$ is the entailment sense set of S_{ij}. The expansion sense set of s_{ij}, $ESS_{ij} = hyperSS_{ij} \cup troponSS_{ij} \cup entailSS_{ij}$. The expansion sense set of W_i, $ESS(w_i) = SS_i \cup hyperSS_{ij} \cup troponSS_{ij} \cup entailSS_{ij}$, in which $hyperSS_i = hyperSS_{i1} \cup hyperSS_{i2} \cup \ldots \cup hyperSS_{iNi} = \{h1S_{i1}, h1S_{i2}, \ldots, h1S_{iM1}\}$, $troponSS_i = troponSS_{i1} \cup troponSS_{i2} \cup \ldots \cup troponSS_{iNi}$, $entailSS_{ij} = entailSS_{i1} \cup entailSS_{i2} \cup \ldots \cup entailSS_{iNi}$.

All the words of each sense $h1S_{ij}$ can be obtained in data.verb and they can be translated into Chinese, thus forming $h1W_{ij}$, the Chinese word set of $h1S_{ij}$. So the Chinese word set of $hyperSS_{ij}$, $hyper(w_i) = h1W_{i1} \cup h1W_{i2} \cup \ldots \cup h1W_{i3}$. The same procedure may be easily adapted to obtain $sysW_i$, $hyponW_i$ and $entailW_i$.

So the verb expansion word set of W_i, $EWS(w_i) = \{sys(w_i), hyper(w_i), tropon(w_i), entailW_i\}$.

3.4 Discover related words based on large-scale corpus

WordNet provides researchers with an effective tool for semantic analysis, but semantic relationship it provides is limited and cannot satisfy users' query requirements sometimes.

For example, when a user submits a query of "*ontology*" to search engine, maybe he wants to know not only "*what is ontology?*", but also the formation and construction rules of ontology, ontology language, or its application. The synset, hypernym and hyponym of ontology can obtained in WordNet, but closely related information to ontology, such as concept, class, relation, function, axioms, instance, RDF, OWL, etc. , cannot be obtained. Closely related information can be found from a large-scale web set.

So this paper discovers related-to relationships among words from large-scale web set and users' search engine performance evaluation data, thus constructing related word set.

3.4.1 Calculate related words based on large-scale web set

(1) Choice of large-scale web set

This paper chooses Sogou full news dataset (SogouCA2012) from Sogou Labs to construct RWS. SogouCA2012 contains news data of eighteen channels from June to July 2012 in several news sites, including national news, international news, sports, society, entertainment, etc. The information of each page in SogouCA2012 includes URL, docID, title and content text. The size of original SogouCA2012 is 711MB. Unzipped SogouCA2012 includes 384 text files and has 2.08GB size.

(2) Process the web page

The front 51 text files in SogouCA2012 are selected. There are 209578 pages left after deleting duplicate and void pages. The title of each page can be found in "<*contenttitle*>" filed. The "<*content*>" field of each page has removed html tag already and retained only body text, so content can be found in "<*content*>" filed.

ICTCLAS is applied to segment title and content of each page. Word frequency is calculated then. Here, only noun, verb are concerned, and preposition, conjunction, pronoun and interjection are ignored. At last, title and content of each page are sorted separately in ascending order of word frequency.

Assume the web set is $DOC = \{d_i = (title_i, content_i) \mid i = 1, 2, \ldots N_1\}$, in which $title_i$ is the title of page d_i and $content_i$ is the content of page d_i, N_1 is 209578. There is a word set $T = \{t_i \mid i = 1, 2, \ldots M\}$. The algorithm for word frequency statistics is as follows.

After word segmentation and word frequency statistics, d_i is expressed as a word list, that is $d_i = \{t_{i1}: n_1, t_{i2}: n_2, \ldots\}$, in which $tt_{ij} \in T$ and n_j is the word frequency of t_{ij} in d_i, $n_j \geq n_{j+1}$, $1 \geq i \geq N$, $1 \geq j \geq M$. d_i can be expressed in $\{t_{i1}: p_1, t_{i2}: p_2, \ldots\}$, in which $p_i = n_i / \Sigma n_i$, after normalized.

(3) Calculate related words
Here words that appear in the same page are considered as related words. The time that word t_i and word t_j appear in the same page is defined as word relevancy $co1_{i,j}$.

Related word list is built in incremental mode. New related words t_i are added to the related word list of current word t_i if they don't exist in t_i's list; otherwise, update t_i's list and the corresponding word relevancy.

The algorithm for related words extraction on the full web set is shown below:

create $tCorList1$ [M], an array of related word list;
for each $d_i \in$ Doc:
for each t_i in d_i:
for each t_i in d_i ($j \neq i$) :
if t_i is in $tCorList_i$
update $co1_{i,j}$++;
else
add t_i to $tCorList_i$;
set $co1_{i,j}$ to1;
end
for each $tCorList_i$:
$co1_{i,j} = co1_{i,j} / N_1$;
sort $tCorList_i$ in ascending order of $co1_{i,j}$;
remove the last third of related words in $tCorList_i$.

Each word t_j in $tCorList1_i$ is a related word of t_i based on web set, and their word relevancy $co1_{i,j}$ is recorded, too. The last third of related words in $tCorList1_i$ are removed because of low relevancy.

3.4.2 Obtain related words based on users' search engine performance evaluation data

Users' search engine performance evaluation data directly show the correlation of user query and result page, so related words can be found from these data. This paper achieves user query and the corresponding related page from SogouE2012 and discovers related words based on it. There are N_2 = 4326 lines in SogouE2012, and each line is in the form of "[*query*]\t *related URL*\t *query type*", in which type 1 means navigation query and type 2 means information query.

The algorithm for related words extraction on search engine performance evaluation data is shown below:

create $tCorList2$ [M], an array of related word list;
for each line in SogouE2012:
extract query word t_i;
fetch pages according to URL using jsoup and get the $title_i$ and $content_i$ of page d_i;
segment $title_i$ and $content_i$, count word frequency and normalize it, then get $d_i = \{t_{i1}: p_1, t_{i2}: p_2, \ldots\}$;
for each d_i:
for each t_i in d_i:
for each t_i in d_i ($j \neq i$) :
if t_i is in $tCorList2_i$
update $co2_{i,j}$++;
else
add t_i to $tCorList2_i$;
set $co2_{i,j}$ to1;
end
for each $tCorList2_i$:
$co2_{i,j} = co2_{i,j} / N_2$;
sort tCorList1$_i$ in ascending order of $co1_{i,j}$.

Each word t_i in $tCorList2_i$ is a related word of t_i based on search engine performance evaluation data, and their word relevancy $co2_{i,j}$ is recorded, too.

3.4.3 Merge two related word list

Above $tCorList2$ and $tCorList2$ are merged into tCorList to facilitate following processing here. Supposing the weight of web set is α_1 and the weight of search engine performance evaluation is α_2, the merge algorithm is as follows.

create related word list of t_i, $tCorList_i$;
add all t_i in $tCorList2_i$ into $tCorList_i$ and set $co_{i,j} = \alpha_2 * co2_{i,j}$
for each t_k in $tCorList1_i$:
if t_k exists in $tCorList_i$:
update $co_{i,k}\ x = \alpha_1 * co1_{i,j}$;
else: add t_k into $tCorList_i$;
$co_{i,k} = \alpha_1 * co1_{i,k}$;
sort $tCorList_i$ in ascending order of $co_{i,j}$.

α_2 is set as 2 times of α_1 here because users' search engine performance evaluation data directly show the correlation of user query and result page. The related words of t_i can be obtained in $tCorList_i$ direct after merging.

3.5 Multi-strategy query expansion method based on semantics

A multi-strategy query expansion method based on semantics is proposed on the basis of section 3.4. In this method, user query is preprocessed first. The semantic

structure of user query is analyzed then. According to the type of user query, specific algorithm is called to get expansion words. The degree of correlation between each expansion word and user query is described in "*req*", which is calculated differently in each detailed algorithm. The overall process is as follows.

acquire the user query;
if query needs splitting:
split it into $q_1, q_2, \ldots, q_L$;
for each q_i:
if q_i needs rewriting:
rewrite q_i according to rewriting rules;
end
else if query needs rewriting:
rewrite query according to rewriting rules;
form the sub-query set of user query: $qset = \{q_1, q_2, \ldots, q_L\}$;
// $L = 1$ if query hasn't been split
for each q_i in qset:

analyze q_i, get the semantic structure expression;

if q_i is of N type:
call the noun expansion method and get the expansion word set eqw_i;

if q_i is of V type:
call the verb expansion method and get the expansion word set eqw_i;

if q_i is of J type:
call the J expansion method and get the expansion word set eqw_i;
end

merge each eqw_i according to req_i, and get eqw, the expansion word set of query;

sort eqw in ascending order of req_i;

get the front M_1 words in eqw as the final expansion words.

3.5.1 The expansion algorithm for noun query

The expansion word of noun query is acquired from web set, search engine performance evaluation data and WordNet. Noun query can be divided into simple noun query and noun phrase query further, and the former is the basis of the latter.

(1) Expansion of simple noun query

The basic idea for simple noun query expansion is to get candidate expansion words from the related word list of q_i at first. Further, increase req of a candidate expansion word if it has semantic relationship with q_i. Supposing the weight of synonymy is β_1, the weight of other semantic relationship is β_2, and the multiple of weight synonymy weight is γ. The expansion algorithm for simple noun query is as follows:

if q_i is of simple N type:
construct eqw_i, the expansion word set of q_i;
get expansion sense set of q_i, ESS $(q_i) = \{$SS (q_i), hyper (q_i), hypon (q_i), meron (q_i), holon $(q_i)\}$, from WordNet;
get sys (q_i), the synonymy set of q_i, from WordNet;
put all the words in q_i's $tCorList_i$ into eqw_i and set $req_{i,l} = co_{i,j}$;
for each t_l in eqw_i:
obtain sense set $SS(t_l) = \{s_{l1}, s_{l2}, \ldots\}$ from WordNet;
for each s_{lm} in $SS(t_l)$:
if s_{lm} is in $SS(q_i)$:
update $req_{i,j}$ += β_1;
else if s_{lm} is in $hyper(q_i)$, $hypon(q_i)$, $meron(q_i)$ or $holon(q_i)$:
update $req_{i,l}$ += β_2;
for each t_n in sys (q_i):
if t_n exists in eqw_i :
update $req_{i,k}$ *= γ;
sort eqw_i in ascending order of $req_{i,j}$.

Synonymy is usually closer to original query than other semantic relationships, so β_1 is set as double the β_2, and β is 2.

(2) Expansion of noun phrase query

The formats of noun phrase queries include "N_1's N_2", "N_1 of N_2", "N_1 and N_2", "N_1 N_2", etc. The expansion word set of each N_j, $eqw(N_j)$, could be acquired according to above simple noun expansion algorithm. All $eqw(N_j)$ are merged into eqw_i, the final expansion word set of q_i. The principle of merging is as follows.

If W_j appears in two expansion word sets, $req_{i,j}$ is the sum of two relevancy. If W_j appears in one expansion word set, $req_{i,j}$ is equal to the corresponding relevancy.

After sorting, the front M_1 expansion words form noun phrase expansion word set of q_i.

3.5.2 The expansion algorithm for verb query

The expansion word of verb query is acquired from web set, search engine performance evaluation data and WordNet. Supposing the weight of synonymy and entailment is β_1, the weight of other semantic relationship is β_2, and the multiple of weight synonymy weight is β.

The expansion algorithm for verb query is as follows:

if q_i is of V type:
extract core verb v_i of q_i;
construct eqw_i, the expansion word set of v_i;
get expansion sense set of v_i, $ESS(v_i) = \{SS(v_i)$, hyper (v_i), tropon (v_i), entail $(v_i)\}$, from WordNet;

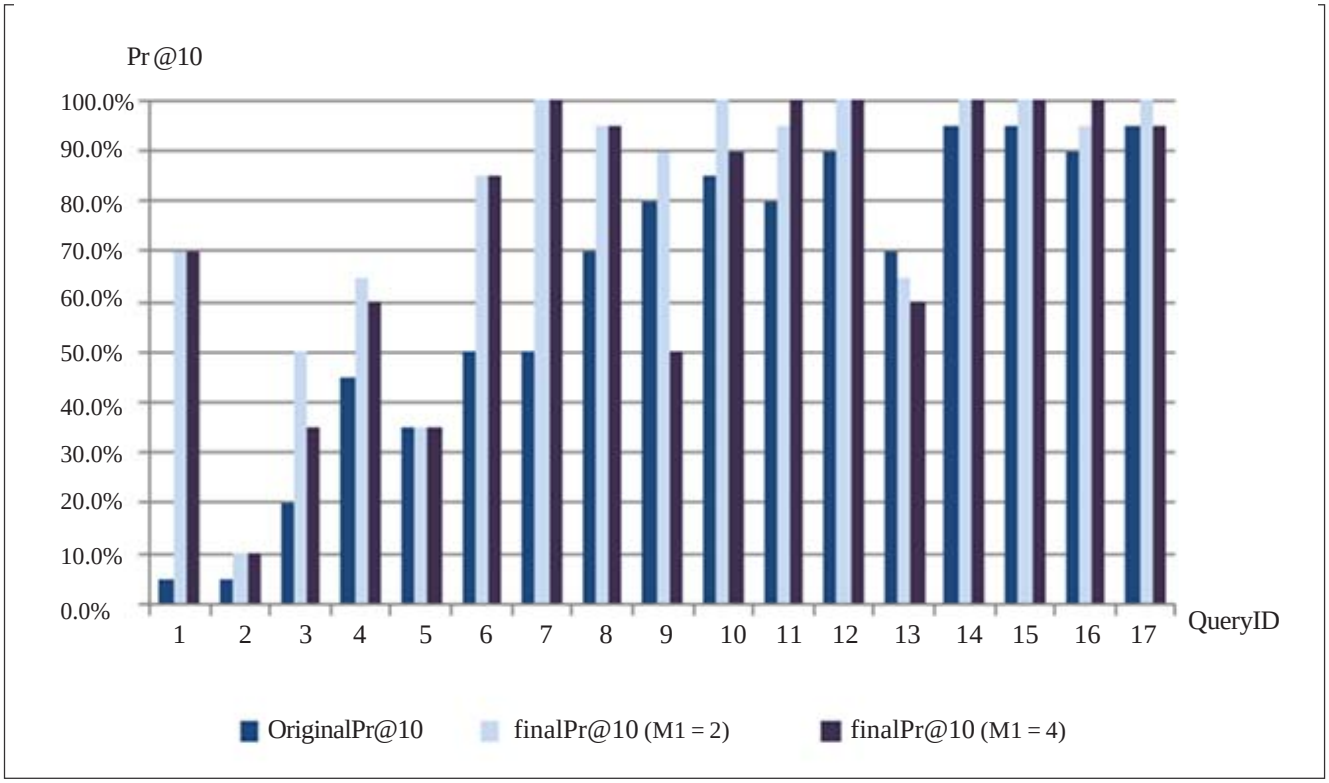

Figure 2. Pr@10 of Each Query

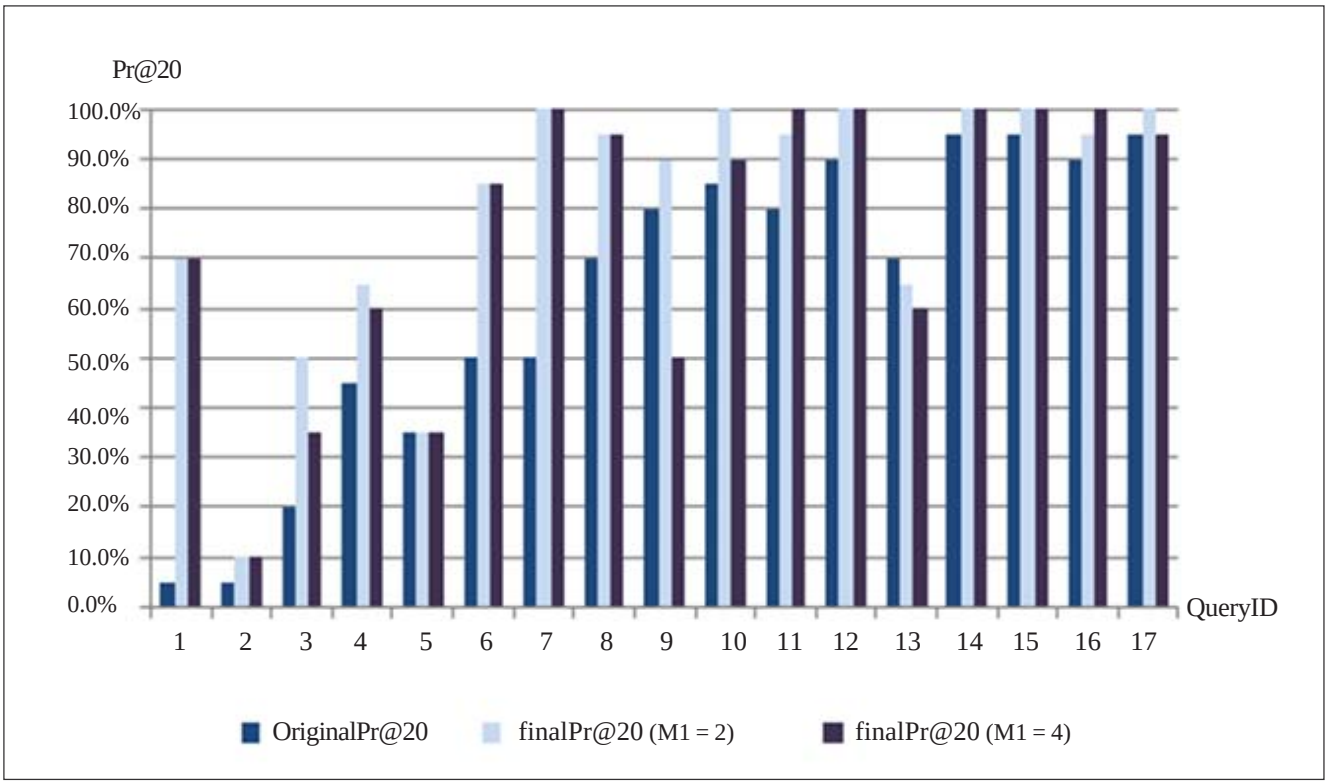

Figure 3. Pr@20 of Each Query

get sys (v_i), the synonymy set of v_i, from WordNet;
put all the words in v_i's $tCorList_i$ into eqw_i and set $req_{i,l} = co_{i,j}$;
for each t_l in eqw_i:
obtain sense set $SS(t_1) = \{s_{l1}, s_{l2}, \ldots\}$ from WordNet;
for each s_{lm} in $SS(v_i)$ or entail (v_i):
if s_{lm} is in $SS(q_i)$:
update $req_{i,l}$ += β_1;
else if s_{lm} is in *hyper* (v_i) or *hypon* (v_i):
update $req_{i,l}$ += β_2;

for each t_n in sys (v_i):

if t_n exists in eqw_i :

update $req_{i,k}$ *= γ;

sort eqw_i in ascending order of $req_{i,j}$;

if q_i has an object:

extract v_i, the object of q_i;

get eqw (o_i), the expansion word set of o_i, according to section 3.5.1;

merge eqw (o_i) and eqw (v_i) into eqw_i, the expansion word set, and sort eqw_i.

Like noun query expansion, synonymy is usually closer to original query than other semantic relationships; so β_1 is set as double the β_2, and γ is 2. The front M_1 words in eqw_i form the expansion word set of q_i.

3.5.3 The expansion algorithm for sentence query

If the user query q_i is a sentence, it needs analyzing further. When q_i is a declarative sentence, the agent and core verb of q_i can be extracted according to its semantic structure expression. The agent and core verb are expanded separately and then merged into the final expansion word set. When q_i is an interrogative sentence, the questioned part of q_i should be analyzed first, according to which q_i is rewritten and then expanded. The expansion algorithm for sentence query is as follows:

if q_i is of J type:

if q_i is a declarative sentence:

get the agent s_i from semantic structure expression of q_i ;

get eqw (s_i), the expansion word set of s_i, according to section 3.5.1;

get the core verb v_i from semantic structure expression of q_i;

get eqw (v_i), the expansion word set of v_i, according to section 3.5.2;

merge eqw (s_i) and eqw (v_i) into eqw_i, the final expansion word set, and sort it;

if q_i is an interrogative sentence:

rewrite it according to the questioned part of q_i; expand it according to setction 3.5.1, and get eqw_i, the expansion word set of q_i.

Similarly, the front M_1 words in eqw_i form the final expansion word set of q_i.

3.5.4 Merge each eqwi into the expansion word set of query

If the query consists of sub-queries q_1, q_2,..., q_L, the expansion word set of each q_i , eqw_i, is acquired based on above algorithm. All the eqw_i are merged into eqw, the expansion word set of query. After sorting, the front M_1 words in eqw form the expansion word set of query.

4. Experimental results and analysis

Thirty user queries are selected r at random from query log (SogouQ.mini2012) distributed by Sogou Labs. Then repeated or illegal queries are deleted. Queries that have high original precision are deleted, too. There are seventeen user queries left at last. These queries are expanded based on above algorithm. Expanded queries are submitted to Sogou Search Engine to verify effectiveness. Supposing the number of expansion word number is M_1, Pr@10 and Pr@20 of each query are counted manually. The experimental results are shown in Figure 2 and Figure 3.

(1) The precision of most queries have improved to some extent after expansion. Among 17 queries, 15 queries get better results than original results. When M1 is 2, the average Pr@10 and Pr@20 have increased by 17.65% and 17.35% respectively. When M1 is 4, the average Pr@10 and Pr@20 have increased by 15.88% and 13.24% respectively.

(2) Adding two expansion words gets better result than adding four expansion words, which means the number of expansion word is not the more the better. In Query 3 and 4, Pr@10 and Pr@20 decreases instead, because original queries are relatively in general terms, and added expansion words result in query drift although they are related to original queries.

(3) Queries that have been rewritten get better results, including queries of "$N + V$" and of interrogative sentence. For example, query 1, 3 and 7. The reason is that rewritten query has more accurate semantic representation, which helps subsequent expansion.

On the other hand, some queries in the experiment are too short to be understood, such as a person, a film, or an organization. It's difficult to learn the true retrieval intention of these queries, so the original precision is pretty high, and the improvement is limited. User feedback can be introduced into these queries.

5. Conclusion

On the basis of analyzing query features, this paper proposed a multi-strategy query expansion method based on semantics. In this method, noun and verb expansion word set are constructed based on WordNet first; then two related word set are constructed based on large-scale web set and users' search engine performance evaluation data respectively, which are merged later. When user submits a query, its semantic structure are analyzed first, according to the type of which different strategy are chosen. Experiments show the effectiveness of this method at last.

6. Acknowledgments

The work presented in this paper is supported by the Fundamental Research Funds for the Central Universities (No. FRF-TP-12-078A and No. FRF-SD-13-001B) and 2014 Open Fund of Beijing Key Laboratory of Knowledge Engineering for Materials Science.

References

[1] Deerwester, Scott., Dumais, Susan T., et al. (2000). Indexing by latent semantic analysis. *ACM Transactions on Information Systems,* 18 (1) 79~112.

[2] Rocchio, Jr, J. J. (1971). Relevance feedback in information retrieval, The SMART Retrieval System: Experiments in Automatic Document Processing, p. 313-323.

[3] Xu, Jinxi., Croft, W. Bruce. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18 (1) 79-112.

[4] Li, Wei-jiang., Zhao,Tie-jun., et al. (2010). Context-sensitive query expansion. *Journal of Computer Research and Development*, 47 (2) 300-304. (In Chinese).

[5] Richardson, R., Smeaton AF. (1995). Using wordnet in a knowledge-based approach to information retrieval. Working Paper CA-0395, Dublin City University.

[6] Cui, Hang., Wen, Ji-rong., et al. (2003). A statistical query expansion model based on query logs. *Journal of Software*, 14 (9) 1593-1599. (In Chinese).

[7] Li, Po-han., He,Zhen-ying., et al. (2011). A Linkage clustering based query expansion algorithm, *Journal of Computer Research and Development,* 48 (S3) 197-204. (In Chinese).

[8] Wu, Yan., Zhang, Qi., et al. (2013). Selecting expansion terms as a set via integer linear programming. *Journal of Computer Research and Development*, 50 (8) 1737-1743. (In Chinese).

[9] Gao, Qing-shi., Hu, Yue., et al. (2003). Semantic language and multi-language MT approach based on SL. *Journal of Computer Science and Technology*, 18 (6) 848-852.

[10] Gao, Qing-shi., Gao, Xiao-yu., et al. (2009). Foundation of unified linguistics. Beijing: Science Press. (In Chinese).

[11] Gao, Qing-shi., Gao, Xiao-yu., et al. (2005). High-speed multi-language machine translation method based on pruning on the tree of representations of semantic elements. *Journal of Software*, 16 (11) 1909-1919.

[12] Hu, Yue., Gao, Xiao-yu., et al. (2008). Formation method of a high-quality semantic unit base for a multi-language machine translation system. *Journal of University of Science and Technology Beijing*, 30 (6) 698-704. (In Chinese).