Search results optimization approach based on semantics

Li Li [1,2]*, Qiang Yao [3]

[1] *School of Computer and Communication Engineering, University of Science and Technology Beijing, No.30 Xueyuan Road, Haidian District, Beijing, China*

[2] *Beijing Key Laboratory of Knowledge Engineering for Materials Science, No.30 Xueyuan Road, Haidian District, Beijing, China*

[3] *Lenovo Corporate Research & Development, Beijing, No.6 Shangdi West Road, Haidian District, Beijing, China*

Received 1 December 2014, www.cmnt.lv

Abstract

Search results optimization technology studies how to process and optimize the original search results so as to improve the user experience. This paper proposed a search results optimization approach based on semantics. This approach represented user query as its semantic structure expression, and represented each original result page as a word list. A correlation calculation model was constructed by combining WordNet, large-scale corpus and users' evaluation data, based on which the correlation between each result page and user query was calculated. When calculating correlation, different module was adopted according to the type of user query. This approach forms a new results list at last. The experiment showed this method can improve retrieval results to some extent.

Keywords: search results optimization, information retrieval, semantic analysis

1 Introduction

Search engine is one of the most used web applications. Search engine returns thousands of results, but users usually view the first two pages on average[1,2]. Users can hardly get useful results if they are not at the top of results list. Search results optimization technology studies how to process and optimize the original search results so as to improve the user experience.

Current search results optimization methods include search result clustering, multiple search results aggregating, machine learning, etc. Search result clustering classifies the original results and displays them in the form of hierarchical structure; users can select different categories based on their interest [3-8]. Search results aggregating, which is usually used in meta search engine, includes simple enumeration, aggregating based on correlation and location, aggregating based on training set, aggregating based on user interest, etc[9-11]. Machine learning method learns query intention rules from user behavior, based on which irrelevant results are filtered [12]. Sharma A.K updates the PageRank of result pages according to the sequential pattern mined from query log, and forms new results list to improve the retrieval effect[13].

Researchers study post-process methods from the point of view of clustering, aggregating, machine learning and data mining, but seldom study on the semantic level. Polysemy and synonymy in natural language impact the search results, because most search engines are based on keyword matching. This paper analysed the query feature in search engine, and proposed a search results optimization approach based on semantics. The correlation between a result page and user query is calculated according to correlation calculation model, which is built based on WordNet, massive web page set and search engine performance evaluation data.

2 Analysis of query feature in search engine

By analyzing real example query log(SogouQ2008) distributed by Sogou Labs, the authors summarized several features of original query[14].

(i) Queries are often expressed in natural languages.

(ii) 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").

(iii) In content, queries are often persons(such as "Fangfei LIU"), places(such as "Wuhu FANGTE"), organizations(such as "Zhuhai Health Center"), products or its attributes(such as "LaCrosse price"), or events (such as "2008 earthquake rescue show").

(iv) 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 when analyzing the correlation between a result page and user query.

* Corresponding author's E-mail: liliustb@126.com

3 Search results optimization approach based on semantics

Several facts are considered as follows when analysing the correlation between a result page and user query:

(i) The semantic structure of user query should be analyzed and understood first. Semantic language[15] is adopted to analyze and express user query.

(ii) Most content of user queries are nouns, verbs and simple sentences. The kernel of a sentence is verb[16], too. So this paper focuses on the analyzing noun and verb, and constructs the correlation calculation model mainly for noun and verb.

(iii) WordNet describes the semantic correlation for each word, which is used as part of the model.

(iv) 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. This paper finds semantic correlation between words from a large-scale web set, thus constructing related words set based on massive web page set.

(v) 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 words set based on search engine performance evaluation data.

Based on the above facts, the overall process of search results optimization based on semantics is shown in Figure 1.

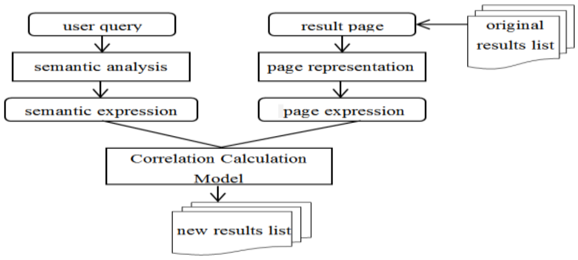

FIGURE 1 The overall process of search results optimization

3.1 SEMANTIC ANALYSIS OF USER QUERY

User query should be analyzed and represented as accurately as possible. The analysis and representation of user query is based on semantic language[15]. 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 [17].

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 [18].

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 and 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, "Where are volcanos located in the world?", "What are earthquake precursors?" , "Yao Ming beat Kobe". Sentences of type J are usually declarative sentences and interrogative sentences. The semantic structure of an interrogative sentence should be rewritten according to its questioned part, for example, "$N:(o_f(V_{olcano},L_{ocation}))$", "$N:(o_f(E_{arthquake}, P_{recursor}))$", etc. A query of interrogative sentence is usually rewritten as a noun phrase.

3.2 RESULT PAGE REPRESENTATION

Search engine returns a result page list to users when they submit a query. The title, abstract and URL of each page are shown in the result list. The title usually summarizes the main content of page. Abstract in result page, which is extracted using automatic abstracting technology, is closely related to user query usually. So this paper focused on the title and abstract of each result page when calculating the correlation. Each result page is expressed in a word list.

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

Assume the result list is $RS=\{d_i=(title_i, abstract_i)|i=1,2,\ldots N_{RS}\}$, in which $title_i$ is the title of result page d_i and $abstract_i$ is the abstract of d_i, N_{RS} is 20. It means the top 20 result pages are analyzed in this paper.

There is a word set $T=\{t_i|i=1,2,\ldots M\}$. 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 $t_{ij} \in T$ and n_j is the word frequency of t_{ij} in d_i, $n_j \geq n_{j+1}$, $1\leq i\leq N, 1\leq j\leq M$. d_i can be expressed in $\{t_{i1}:p_1,t_{i2}:p_2,\ldots\}$, in which $p_i= n_i/\sum n_i$, after normalized.

3.3 CORRELATION CALCULATION MODEL FOR SEARCH RESULTS OPTIMIZATION

The correlation calculation model between result pages and user query consists of four parts[14], which is shown in Figure 2.

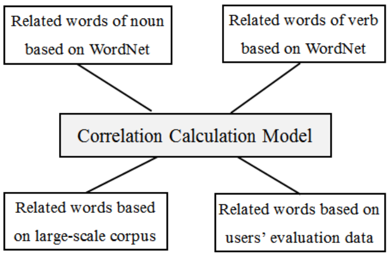
Related words of noun based on WordNet
Related words of verb based on WordNet
Correlation Calculation Model
Related words based on large-scale corpus
Related words based on users' evaluation data

FIGURE 2 Correlation calculation model for search results optimization

3.3.1 Related words of noun based on WordNet

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

(i) Translation between English words and Chinese words

WordNet is built in English, but this paper mainly process Chinese web page, so translation between English and Chinese 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.

(ii) Form noun related word set

There are 82192 noun synset and 117953 nouns in WordNet3.1. 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, $SS_i=\{s_{i1},s_{i2},\dots s_{iNi}\}$, can be obtained in noun index file (index.noun). Here S_{ij} is the j-th sense of W_i, and it is expressed in the sense_number in WordNet. RSS_i, the related 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 calculation. Suppose $hyperSS_{ij}$ 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 related sense set of s_{ij}, $RSS_{ij} = hyperSS_{ij} \cup hyponSS_{ij} \cup meronSS_{ij} \cup holonSS_{ij}$. The related sense set of W_i, $RSS_i =hyperSS_{ij} \cup hyponSS_{ij} \cup meronSS_{ij} \cup holonSS_{ij}$, in which $hyperSS_i = hyperSS_{i1} \cup hyperSS_{i2} \cup \dots \cup hyperSS_{iNi}=\{h1S_{i1}, h1S_{i2}, \dots, h1S_{iM1}\}$, $hyponSS_i = hyponSS_{i1} \cup hyponSS_{i2} \cup \dots \cup hyponSS_{iNi}$, $meronSS_i= meronSS_{i1} \cup meronSS_{i2} \cup \dots \cup meronSS_{iNi}$, $holonSS_i= holonSS_{i1} \cup holonSS_{i2} \cup \dots \cup holonSS_{iNi}$.

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 $hyperSS_{ij}$, $hyper(w_i)= h1W_{i1} \cup h1W_{i2} \cup \dots \cup h1W_{i3}$. The same procedure may be easily adapted to obtain $sys(w_i)$, $hypon(w_i)$, $meron(w_i)$ and $holon(w_i)$.

So the synonyms of noun W_i is $sys(w_i)$, and the noun related word set of W_i, $RWS(w_i)=hyper(w_i) \cup hypon(w_i) \cup meron(w_i) \cup holon(w_i)$.

3.3.2 Form verb related word set 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, $SS_i=\{s_{i1},s_{i2},\dots s_{iNi}\}$, can be obtained in verb index.verb. Here S_{ij} is the jth sense of W_i, and it is expressed in the sense_number in WordNet.

RSS_i, the related 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 calculation. 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 related sense set of s_{ij}, $RSS_{ij} = hyperSS_{ij} \cup troponSS_{ij} \cup entailSS_{ij}$. The related sense set of W_i, $RSS(w_i) = hyperSS_{ij} \cup hyponSS_{ij} \cup entailSS_{ij}$, in which $hyperSS_i = hyperSS_{i1} \cup hyperSS_{i2} \cup \dots \cup hyperSS_{iNi} = \{h1S_{i1}, h1S_{i2}, \dots, h1S_{iM1}\}$, $troponSS_i = troponSS_{i1} \cup troponSS_{i2} \cup \dots \cup troponSS_{iNi}$, $entailSS_{ij} = entailSS_{i1} \cup entailSS_{i2} \cup \dots \cup entailSS_{iNi}$.

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 $hyperSS_{ij}$, $hyper(w_i)= h1W_{i1} \cup h1W_{i2} \cup \dots \cup h1W_{i3}$. The same procedure may be easily adapted to obtain $sys(w_i)$, $tropon(w_i)$ and $entail(w_i)$.

So the synonyms of verb W_i is $sys(w_i)$, and the verb related word set of W_i, $RWS(w_i)= hyper(w_i) \cup tropon(w_i) \cup entail(w_i)$.

3.3.3 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. 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, thus constructing related word set.

Sogou full news dataset (SogouCA2012) from Sogou Labs is selected to discover related words. The front 51 text files in SogouCA2012 are used here. 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 already removed html tag and retained only body text, so content can be found in "<content>" filed.

Each page can be expressed in the same way as section3.2. Assume the web set is $DOC=\{d_i= (title_i, content_i)|i=1,2,\dots N_1\}$, 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. Page d_i can be expressed in $\{t_{i1}:p_1, t_{i2}:p_2, \ldots\}$, in which $p_i = n_i/\sum n_i$.

Words that appear in the same page are considered as related words here. 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_j 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 $t_j$ in $d_i$ $(j \neq i)$ :
            if $t_j$ is in tCorList$_i$
                update $co1_{i,j}$++;
            else
                add $t_j$ to tCorList$_i$;
                set $co1_{i,j}$ to 1;
For each tCorList$_i$:
    $co1_{i,j} = co1_{i,j} / N_1$;
    sort tCorList1$_i$ in ascending order of $co1_{i,j}$;
    remove the second half of related words in tCorList1$_i$.
```

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 second half of related words in tCorList1$_i$ is removed because of low relevancy.

3.3.4 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 $t_j$ in $d_i$ $(j \neq i)$ :
            if $t_j$ is in tCorList2$_i$
                update $co2_{i,j}$++;
            else
                add $t_j$ to tCorList2$_i$;
                set $co2_{i,j}$ to 1;
                End
    For each tCorList2$_i$:
        $co2_{,j} = co2_{i,j} / N_2$;
        sort tCorList1$_i$ in ascending order of $co1_{i,j}$.
```

Each word tj 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.

3.3.5 Merge related word list

Above tCorList1 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 $t_j$ 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 $co_{i,k}$ $+= \alpha_1 * co1_{i,j}$ ;
    else: add $t_k$ into tCorList$_i$;
        $co_{i,k} = \alpha_1 * co1_{i,k}$ ;
sort tCorList$_i$ in ascending order of $co_{i,j}$;
```

Here α_2 is set as 2 times of α_1 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.

3.4 SEARCH RESULTS OPTIMIZATION BASED ON SEMANTICS

A search results optimization approach based on semantics is proposed on the basis of section 3.3. User query is submitted to search engine and original result page list is acquired. On the other hand, the semantic structure of user query is analyzed to get its semantic structure expression. Specific algorithm is called to calculate the correlation between a result page and user query according to the type of user query. At last, new result page list is formed in ascending order of correlation. The overall process is as follows.

```
acquire the user query;
submit the query to search engine and get original results list $RD=\{Rd_1, Rd_2, \ldots, Rd_{20}\}$;
for each $Rd_j$ in RD：
    represent $Rd_j$ as $Rd_j=\{ t_{j1}:p_1, t_{j2}:p_2, ,t_{j3}:p_3, \ldots\}$;
preprocess query and form a sub-query set of query, $qset=\{q_1, q_2, \ldots, q_L\}$;
for each $q_i$ in qset:
    analyze $q_i$ , get itssemantic structure expression;
    for each $Rd_j$ in RD:
        if $q_i$ is of N-type:
            call the corrlation calculation module of noun query, and get the correlation between $Rd_j$ and $q_i$, $req_{i,j}$ ;
        if $q_i$ is of V-type:
            call the corrlation calculation module of verb
            query, and get the correlation between $Rd_j$ and $q_i$,
            $req_{i,j}$ ;
        if $q_i$ is of J-type:
            call the corrlation calculation module of J query,
            and get the correlation between $Rd_j$ and $q_i$, $req_{i,j}$ ;
    end
end
for each $Rd_j$ in RD:
    calculate the correlation between $Rd_j$ and query, $req_j$
    $=\sum req_{i,j}$ ;
end
sort RD={$Rd_1$, $Rd_2$, ..., $Rd_{20}$} in ascending order of $req_i$ and
form the new results list.
```

3.4.1 Preprocess the user query

Users can input their queries freely in search engine, so some queries needs preprocessing before their sematic structures are analyzed. Two kinds of user queries should be preprocessed:

(i) Queries that are composed of more than one word 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”.

(ii) 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”.

The process of preprocessing is as follows.

```
acquire the user query;
    if query needs splitting:
        split it into $q_1$, $q_2$, ..., $q_L$;
        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 = {$q_1$,$q_2$, ...,$q_L$}.
// L=1 if query hasn’t been split
```

3.4.2 Correlation calculation module of noun query

The correlation between each result page Rd_j={t_{j1}:p_1, t_{j2}:p_2, t_{j3}:p_3,…} and sub-query q_i of noun type, $req_{i,j}$, is calculated on the basis of synonymy based on WordNet, related words based on WordNet and related words based on large-scale corpus and users’ evaluation data. Supposing the weights are α, β and γ respectively.

Noun query can be divided into simple noun query and noun phrase query further, and the former is the basis of the latter.

(i) Correlation calculation algorithm of simple noun query

A simple noun query consists of a single noun. The Correlation between Rd_j and simple noun query q_i can be translated into the correlation between each word t_{jk} in Rd_j and q_i. On the other hand, the importance of each t_{jk} in Rd_j is different, which has a lot to do with p_{jk}.

Supposing $req_{i,j}$ is the correlation between Rd_j and simple noun query q_i, $req_{i,jk}$ is the correlation between word t_{jk} in Rd_j and q_i, $req_{i,j}$ is acquired according to the following algorithm.

```
if $q_i$ is N-type:
    set $req_{i,j} = 0$, each $req_{i,jk}=0$;
    for each $t_{jk}$ in $Rd_j$:
        get sys($q_i$), the synonymy set of $q_i$ from WordNet;
        get RWS($q_i$), the related word set of $q_i$ from
        WordNet;
        if $t_{jk}$ is in sys($q_i$):
            $req_{i,jk}$+=α;
        if $t_{jk}$ is in RWS($q_i$):
            $req_{i,jk}$+=β;
        get tCorList($q_i$), the related word list of $q_i$;
            if $t_{jk}$ is in tCorList($q_i$):
                $req_{i,jk}$+=γ*$co_{i,jk}$;
            $req_{i,jk}$*=$q_{jk}$;
        end
        $req_{i,j}=\sum req_{i,jk}$ .
    end
```

Synonymy is usually closer to original query than other semantic relationships; Related words extracted from large-scale corpus and users’ evaluation data show the statistical correlation; so α is set as double the β and γ here.

(ii) Correlation calculation of noun phrase query

According to section 2, the formats of noun phrase queries are “N_1’s N_2”, “N_1 of N_2”, “N_1 and N_2”, “N_1 N_2”, etc. The correlation between Rd_j and each simple noun N_k can be calculated using above algorithm respectively, and then added up to get $req_{i,j}$.

3.4.3 Correlation calculation module of verb query

Most of the semantic structure expression of v-type user query is in the form of “main verb v_i [+*object* o_i]” according to section 2, so the correlation between result page Rd_j and q_i can be split into two parts. The correlation calculation module is as followings:

```
if $q_i$ is of V-type:
    extract main verb $v_i$ of $q_i$ from its semantic structure
    expression;
    set $req_{i,j}$=0, each $req_{i,jk}$=0;
    for each $t_{jk}$ in $Rd_j$:
        get sys($v_i$), the synonymy set of $q_i$ from WordNet;

        get RWS($v_i$), the related word set of $q_i$ from
        WordNet;
        if $t_{jk}$ is in sys($v_i$):
            $req_{i,jk}$+=α;
        if $t_{jk}$ is in RWS($v_i$):
            $req_{i,jk}$+=β;
        get tCorList($v_i$), the related word list of $q_i$;
        if $t_{jk}$ is in tCorList($v_i$):
            $req_{i,jk}$+=γ*$co_{i,jk}$;
        $req_{i,jk}$*=$q_{jk}$;
    end
    $req_{i,j}=\sum req_{i,jk}$;
    if $q_i$ has object:
            extract object $o_i$ of $q_i$ from semantic structure
            expression;
          calculate the correlation between $Rd_j$ and $o_i$, and add
          it to $req_{i,j}$;
        end
    end
```

3.4.4 Correlation calculation module of sentence query

If the user query q_i is a sentence, it needs analysing further. When q_i is a declarative sentence, the agent and main verb of q_i can be extracted according to its semantic structure expression. When qi is an interrogative sentence, the questioned part of q_i should be analysed first, according to which q_i is rewritten and then calculated. The correlation calculation algorithm of sentence query is as followings:

```
if $q_i$ is of J-type:
    if $q_i$ is a declarative sentence:
        set $req_{i,j}$=0, each $req_{i,jk}$=0;
      for each $t_{jk}$ in $Rd_j$:
            get the agent $s_i$ from semantic structure
            expression of $q_i$;
            calculate the correlation between $Rd_j$ and $s_i$,
            $req1_{i,j}$ according to section 3.4.2;
            get the main verb $v_i$ from semantic structure
            expression of $q_i$;
            calculate the correlation between $Rd_j$ and $v_i$,
            $req2_{i,j}$,according to section 3.4.3;
            correlation between $Rd_j$ and $q_i$, $req_{i,j}$=
            $req1_{i,j}$+$req2_{i,j}$ ;
    end
    if $q_i$ is an interrogative sentence:
            rewrite it according to the questioned part of
            $q_i$;
          calculate correlation between $Rd_j$ and $q_i$, $req_{i,j}$,
          according to setction 3.4.2.
    end
end
```

3.4.5 Form the new results list

If user query consists of multiple sub-queries $q_1,q_2,\cdots,q_L$, all $req_{i,j}$ are added up to form correlation between Rd_j and user query, $req_j=\Sigma\, req_{i,j}$ $(i=1,2,\cdots,L)$.

Then the original result list RD={Rd_1, Rd_2, …, Rd_{20}} is sorted in ascending order of each req_i and form the new results list.

4 Experimental results and analysis

Ten user queries are selected from query log (SogouQ.mini2012) distributed by Sogou Labs. The format of query log is "[AscessTime\t SessionID\t QueryTerm\t Rank\t SequenceNumber\t URL]". Repeated or illegal queries and queries whose SequenceNumber(SN) is 1 are ignored so as to verify the effect of above approach.

Each query is submitted to Sogou search engine, and the original results list is returned. This paper records the original Sequence Number (original SN), and count the precision of top ten results(Original Pr@10).

Then the above approach is adopted to calculate the correlation between each result page and user query, and new results list is formed. This paper records the new Sequence Number(Improved SN), and count the precision of top ten results(Improved Pr@10) again. The experimental results are shown in Figure 3 and Figure 4.

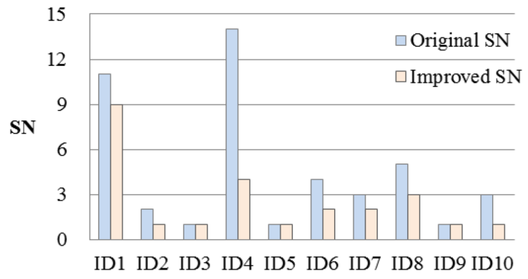

FIGURE 3 Comparison of Sequence Number of each query

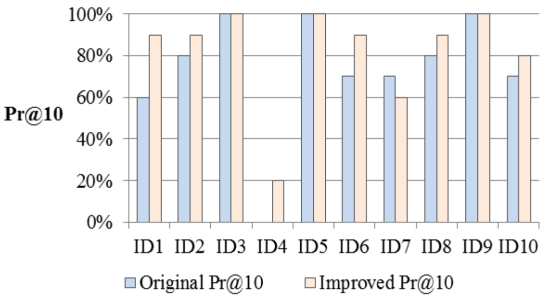

FIGURE 4 Comparison of Pr@10 of each query

Figure 3 shows the Sequence Numbers have decreased to some extent after optimization. It means the result users need really is ranked at the top of the new results list. Among ten queries, seven queries get lower SN than original results, and the average SN decreases by 2. Query 4 gets the best result, because segmentation in search engine causes key information loss and low original precision; while optimization in this paper is based on semantic structure, thus avoiding information loss and leading to lower SN. SNs of Query 3,4 and 9 have not changed because of high original precision.

Figure 4 shows the precision of most queries have improved to some extent after optimization. It also means more results that users need really are ranked at the top of the new results list. Among ten queries, six queries get higher Pr@10 and the average Pr@10 increases by 9%. Pr@10 of Query 3, 4 and 9 haven't changed because the contents of these three queries are specific product or event and they have quite high original precision already. Pr@10 of Query 7 has decreased because the abstract of some result page in original results list is not right, which affect the calculation of correlation.

5 Conclusions

On the basis of analysing query features, this paper proposed a search results optimization approach based on semantics. In this approach, user query is represented as its semantic structure expression, and original result page is represented as a word list. This paper constructs a correlation calculation model on the basis of WordNet, large-scale corpus and users' evaluation data, calculate the correlation between each result page and user query according to the type of user query, and forms a new results list. Experiments show the effectiveness of this approach at last.

Reference

[1] Pierre Baldi, Paolo Frasconi, Padhraic Smyth 2003 *Modeling the Internet and the Web Probabilistic Methods and Algorithms* Wiley: New Jersey

[2] SHAN S W 2003 *search engine, Log Analysis in Search Engine: Method, Technology and Application*. Peking University: Beijing *(in Chinese)*

[3] ZHOU C 2009 *Document Clustering in Search Engine*. Hua Zhong University of Science and Technology: Wuhan *(in Chinese)*

[4] SHEN Y H, FENG X L, HUANG R Y 2010 Search Results Optimization Based on Web Clustering Algorithm. *Journal of Computer Applications* **S1**(30) 51-3 *(in Chinese)*

[5] LIU D S 2011 Improved Search Results Clustering Algorithm Based on Suffix Tree Model *Computer Science* **38**(11) 148-52 *(in Chinese)*

[6] Qin Yanxia, Zheng Dequan, Zhao Tiejun 2012 Research on search results optimization technology with category features integration *International Journal of Machine Learning and Cybernetics* **3**(1) 71-76

[7] Qin Yanxia, Zheng Dequan, Xu Bing 2011 Search results optimization method combined with multi-features *Proc. of Int. Conf. on Fuzzy Systems and Knowledge Discovery(Shanghai)* vol 2 IEEE Computer Society: DC p1167

[8] Bordogna Gloria, Campi Alessandro, Psaila Giuseppe 2012 Disambiguated query suggestions and personalized content-similarity and novelty ranking of clustered results to optimize web searches *Information Processing and Management* **48**(3) 419-37

[9] PENG X H, ZHANG L, YU J Q 2003 Result Optimization of Agent-based Meta Search Engine *Computer Applications* **23**(12) 68-70 *(in Chinese)*

[10] LI Q Q，TAN X C，JIN M X 2011 Research of Result Conformity Algorithm based on Personalized Meta-search *Science Technology and Engineering* **33**(11) 8361-8366 *(in Chinese)*

[11] Li L 2013 *Research on the key technology of the personalized meta-search engine* Inner Mongolia University of Science and Technology: Baotou*(in Chinese)*

[12] Huang L 2008 *Design and Implementation of Case-Based Learning System for Optimizing Results from Search Engine* Nanchang University: Nanchang *(in Chinese)*

[13]Sharma A.K., Aggarwal Neha, Duhan Neelam 2010 Web search result optimization by mining the search engine query logs *Proc. of Int. Conf. on Methods and Models in Computer Science(New Delhi)* IEEE Computer Society: New Jersey p 39

[14] LI LI 2014 Construction of semantic knowledge base for query expansion *WIT Transactions on Information and Communication Technologies* **62** 1129-37

[15] Gao Q S , Hu Y, Gao X Y 2003 Semantic language and multi-language MT approach based on SL *Journal of Computer Science and Technology* **18**(6) 848-52

[16] Qingshi Gao, Xiaoyu Gao 2009 *Foundation of unified linguistics* Science Press: Beijing (*In Chinese)*

[17] Gao X Y, Gao Q S, Hu Y, Li L 2005 High-speed multi-language machine translation method based on pruning on the tree of representations of semantic elements *Journal of Software* **16**(11) 1909-19

[18] Yue Hu, Xiao-yu Gao 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)

Authors

Li Li, 1980.9, Zhumadian City, Henan Province, P.R.China

Current position, grades: lecturer of School of Computer and Communication Engineering, University of Science and Technology Beijing, China
University studies: received her Phd.Sc in Computer Application from University of Science and Technology Beijing in China.
Scientific interest: Her research interest fields include natural language processing and information retrieval.
Publications: 10 articles in various journals and conferences
Experience: She has teaching experience of 6 years.

Yao Qiang, 1980.9, Zhumadian City, Henan Province, P.R.China

Current position: engineer of Lenovo Corporate Research & Development, Beijing
University studies: received his B.Sc in Computer Science and Technology from Beijing University of Posts and Telecommunications in China.
Scientific interest: His research interest fields include algorithm design, human interaction and device innovation.
Experience: He has large-scale project experience of 13 years.