Abstract—In this paper, we propose a re-ranking method which employs semantic similarity to improve the quality of search results. We fetch the top N results returned by search engine, and use semantic similarities between the candidate and the query to re-rank the results. We first convert the ranking position to an importance score for each candidate. Then we combine the semantic similarity score with this initial importance score and finally we get the new ranks. In the experiment, we use NDCG to evaluate the re-ranking results and the experimental results validate that our proposed method can indeed improve the search performance and meet users’ need to a certain extent.

I. INTRODUCTION

As the information on the Internet explodes astonishingly, search engines play a more and more important role. However, due to the diversity of web users’ search requests, it is urgent for search engines to improve their keyword-based search techniques.

Currently search techniques are mainly based on keyword matching. However, this technique has the following weaknesses. First, web users cannot express their search intention accurately using several keywords half the time. Hence the exactly-matched results do not consequentially satisfy the web users. Secondly, keyword matching cannot guarantee the selected candidates have high correlation with the user query, given the different positions and meanings of the keywords. Third, under the circumstances of keyword matching, the top-ranked search results for a given query must contain the keywords as much as possible, otherwise they will lose their ranking positions although their contents exactly discuss the same thing. This will also lead to an awkward situation: spammers try their best to pollute the web document corpus with term spamming tricks such as repetition, dumping and weaving.

Another problem about current search engines is their ranking schemes. PageRank is the most popular ranking algorithm, however, it is based on the popularity of web documents, not the quality. Therefore, a newborn web document usually cannot get highly-ranked positions due to their freshness and thus little reputation. How to promote the new documents and maximize quality of search results seen by users is becoming a more and more challenging work.

In this paper, we try to use semantic analysis method to remedy the shortcomings of the current search techniques mentioned above. We believe the search based on lexical semantics instead of keyword matching can better adapt to the thinking pattern of human beings, and thus search results are more relevant to users’ search intention. Meanwhile, using semantic factors can conciliate the freshness and make the high-relevant new pages get moderate rank promotion.

In our work, we fetch the top N results returned by search engines such as Google for user queries, and use semantic similarities between the candidate and the query to re-rank the results. We first convert the ranking position to an importance score for each candidate. Then we combine the semantic similarity score with this initial importance score and finally we get the new ranks. We analyze the combination ratio between these two parts and choose a best one. The experimental results validate that our proposed method can indeed improve the search performance.

The remainder of this paper is organized as follows. Semantic similarity is introduced in Section 2. Our re-ranking method is discussed in Section 3. Then the experimental results are shown in Section 4. The related works are reviewed in Section 5. Finally we conclude the paper in Section 6.

II. SEMANTIC SIMILARITY

To exploit semantic similarity, an ontology must be specified first. For example, WordNet\(^1\) is a very famous and widely-used ontology. Based on WordNet, researchers have already put forward some semantic similarity formulas. For example, Leacock and Chodorow propose the following formula to compute the semantic between two concepts [7]:

\[
sim_{LC}(\pi_1, \pi_2) = -\log \frac{\max_{\pi \in \text{wordnet}} \text{depth}(\pi)}{\text{len}(\pi_1, \pi_2)}
\]

where \(\text{len}(\pi_1, \pi_2)\) is the length of the shortest path between concept \(\pi_1\) and \(\pi_2\) in WordNet and \(\text{depth}(\pi)\) is the length of the path from \(\pi\) to the root.

The above formula does not consider the positions of the two concepts appeared in the ontology. We know usually the upper-level concepts are of more generalization than lower-level concepts. As noted by Sussna [12], sibling-concepts with larger depth are more likely to have semantic correlations than the higher ones. Based on this observation, we propose the

\(^1\)http://wordnet.princeton.edu
following formula for semantic similarity in this paper,

$$\text{sim}(\pi_1, \pi_2) = \frac{\log \frac{\text{len}(\pi_1, \pi_2)}{\text{depth}(\pi_1) + \text{depth}(\pi_2)}}{\log \frac{1}{2(\max_{\pi \in \text{wordnet}}(\text{depth}(\pi)+1))}}$$

(2)

The ontology we use in this paper is WorkiNet instead of WordNet. WorkiNet, which is a product of our lab, is an enhanced ontology by integrating the information of Wikipedia into WordNet. WorkiNet keeps the desirable structure of WordNet, meanwhile encapsulates abundant information from Wikipedia. The version used in this paper includes 1,782,276 new concepts borrowed from Wikipedia.

III. RE-RANKING METHOD

As introduced in Section I, we use a linear combination of the importance score of semantic similarity score for each candidate. In this Section, we will describe how to get these two scores and how to combine them in detail.

A. Importance

since our algorithm is heavily dependent on the search engine’s quality and result, how to grade the web pages returned from the search engine is important. So we need to find a criteria to measure each web page. Before introducing our method, we would like to show one important concept, the DCG, short for Discounted Cumulative Goal. It is a common way to evaluate the search result quality using DCG.

The expression of DCG is as follows [5].

$$\text{DCG}_p = \sum_{i=1}^{p} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}$$

(3)

where \(p\) is PageRank serial number and \(\text{rel}_i\) is the graded relevance of the result at position \(i\). This expression has a discount factor, which makes the lower the rank, the smaller share of the document value is added to the cumulated gain. The lower the ranked position of a relevant document - of any relevance level - the less valuable it is for the user, because it is less likely that the user will examine the document due to time, effort, and cumulated information from documents that already seen.

Let us go back to how to measure the importance of the results at different positions. As we know, the search results are returned by search engines according to their importance and relevance. The most important web pages usually are returned at the top positions, and hence attract much more attention from users. On the contrary, the unimportant pages are returned at the bottom positions. Therefore, a discount factor is needed which progressively reduces the document value as its rank decreases. An intuitive and simple way for this requirement is also to use a log value as the divisor, just like the DCG does.

We propose the following formula to calculate each web’s importance score.

$$\text{importance}(i) = \frac{1 - (i - 1)/\text{tot}}{\log_2(i+1)}$$

(4)

where \(i\) is the original PageRank serial number (i.e., the original ranking position) and \(\text{tot}\) is the number of the fetched web pages for a query. The formula indicates that the top results have significant importance to the search keywords and thereby are much valuable for web users.

In order to make calculation simple, the result is normalized. For example, if \(i\) equals to 1, the importance of the first web page is 1, which is the highest score. Along with the increasing of the ranking position number of a returned result, its importance score is decreased sharply.

B. Relevance

Relevance is the semantic similarity between keywords and a specified web document. Our algorithm is different from other semantic similarity methods. We calculate the similarities between the keywords and each word in the document (of course removing the stop words) to get the final relevance.

Firstly, we get all the web page content and use segmental tool to segment word because the words like "such", "of", which are called stop words, must be deleted. Then we put the remaining words and the times they appear in each web page into an array and sort this array according to the times in descending order. We only get former fifty percent of the words in the array to calculate the relevance because other words in web pages are totally unrelated. We can understand this method easily by using this formula:

$$\text{sem}(j)_{\text{words}} = \text{sim}(j, k) = \frac{\log \frac{\text{len}(j,k)}{\text{depth}(j) + \text{depth}(k)}}{\log \frac{1}{2(\max_{\pi \in \text{wordnet}}(\text{depth}(\pi)+1))}}$$

(5)

where the \(\text{words} \) is an aggregation of former-fifty-percent words in each web page and \(j\) is a word in web page. The expression \(\text{sim}(a,b)\) is from Section II, the first formula. It’s clear that \(\text{sem}(j)\) is the semantic similarity of \(j\) and the keyword.

After defining the semantic similarity, we will weight each of the word in the web page and calculate the relevance. Then we can get the relevance of each web page by those values of semantic similarity. The formula to compute relevance of each web page is:

$$\text{relevance}(i) = \sum_{j \in \text{words} \prod \text{words} \subseteq \text{web}(i)} \text{sem}(j) \times \text{times}(j)\sum_{j \in \text{words} \prod \text{words} \subseteq \text{web}(i)} \text{times}(j)$$

(6)

Where \(\text{relevance}(i)\) is the relevance of web page \(i\) and \(\text{times}(j)\) means the number of the word \(j\) appeared in web page. The index of the web page is \(\text{web}(i)\). According to WorkiNet introduced above, we know that the value of \(\text{sem}\) is between 0 and 1. So the relevance score of each web page is between 0 and 1.

C. Combination and improvement

So far we have defined the importance and semantic relevance of each web page. We should combine them to get
the final ranking score and use this score to re-rank the search results. In this paper, we adopt a linear combination to keep our solution simple. The adjusting parameter used in the combination is called $\theta$.

$$final(i) = \theta \times importance(i) + (1 - \theta) \times relevance(i) \quad (7)$$

The value of $\theta$ is between 0 and 1, and hence final(i) is between 0 and 1 either.

$$\text{Similarity(Word 1, Word 2, \ldots, Word j, \ldots, Word n, \text{Query})} = \text{Calculate The Similarity Score}$$

$$\text{Importance(Rank i) = \text{Calculate The Importance Score}}$$

$$\text{Mix 2 Kinds of Scores}$$

$$\text{Final Score for i-th Document}$$

Fig. 1. Our re-ranking method

As the framework shown in Fig.1, our re-ranking method has three main steps. However, we have exceptions when dealing with web pages such as PDF, DOC, or PPT documents. It is hard for us to compute their relevance with the search keywords since we cannot get their text contents directly. So we must think of other ways to handle them.

There are two straightforward methods to deal with such documents. We can keep such a document at the same position (unchanged) or just use the importance value as the final re-ranking score. We will discuss which one is better in our experiments. At the same time, we do not consider the semantic relevance for a page if it contains only one or two words, since the relevance is difficult to compute in that scenario.

IV. EXPERIMENT

A. Information about data and resource

Since the importance is based on the search engine and our result heavily relies on the search engine, the quality of search engine directly affects our search result. We know the quality of Google is very good. So in our experiment, we choose Google as our desirous search engine and use the results returned by Google as the re-ranking pool. For each query, we crawl down the top 600 web documents returned by Google and put these documents into the re-ranking pool. Then we process every document and extract their text contents as far as we can. In total we download 120,000 web documents for 200 queries.

B. Evaluation method and Volunteer study

We have several volunteers to assign each document a score according to its relevance to the query. Hence we get an optimal ranking list for the returned results manually. Then we compute the NDCG values of each query for different methods, and average the NDCG values for all queries.

When volunteers conduct the evaluation, they have to obey some principles. Since our semantic similarity computation is based on the values between words, and a word may have different meanings, the volunteer must decide which meaning is appropriate for the keywords. We believe that the original meaning is the best one. For example, when the query is “apple” Google may return documents about the fruit apple, the Apple corporation or any other documents that just contain the word “apple” In this scenario, the fruit apple is chosen as the user’s search intention.

From Table I, we can see that how we do the experiment and volunteers mark the URL. G-rank means the rank given by Google, while S-rank is the result of our algorithm and scores are the final results of importance and relevance in a given proportion. Marked score is the weight of each URL that marked by the volunteers.

<table>
<thead>
<tr>
<th>S-rank</th>
<th>G-rank</th>
<th>URL</th>
<th>Score</th>
<th>Marked</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td><a href="http://www.cancer-fund.org/">http://www.cancer-fund.org/</a></td>
<td>0.42170</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td><a href="http://en.wikipedia.org/wiki/">http://en.wikipedia.org/wiki/</a>..</td>
<td>0.39691</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td><a href="http://www.cancer.gov/">http://www.cancer.gov/</a></td>
<td>0.35846</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td><a href="http://www.hkbcf.org/">http://www.hkbcf.org/</a></td>
<td>0.30172</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td><a href="http://www.cancer.net/">http://www.cancer.net/</a></td>
<td>0.25363</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td><a href="http://www.medicalnewstoday">http://www.medicalnewstoday</a>...</td>
<td>0.24665</td>
<td>5</td>
</tr>
</tbody>
</table>

C. NDCG calculation

NDCG is an effective measure mainly used in information retrieval research to evaluate rankings of documents according to their relevance [6], [3]. It measures how a ranking algorithm is in assigning suitable ranking to relevant documents. For example, if we have three web pages d1, d2, d3 whose relevance scores are (3, 2, 1) respectively (the higher score, the more relevant), then the ranking of (d1, d2, d3) will achieve a higher NDCG value than the ranking of (d3, d2, d1).

In our experiment, we can get the best rank according to volunteers. Then we give this URL weight based on their rank number. The formula to compute weight of each URL is:

\[ rel_i = \left( \frac{tot - i + 1}{tot} \right) \quad (8) \]

where $i$ is page-rank number and $tot$ is the number of web pages. Now URLs have weights and in the experiment they were sorted from different ways, just like the Table 1 shown in the last Section. And from each sorting we can get a value of DCG. We know how to calculate DCG from the expression 3 and now we have to calculate IDCG. IDCG is the ideal DCG.
Firstly, we get the search results. Then by artificially sorting, we have the best rank. And calculate the order of query of DCG.

Now we can compute NDCG of each rank:

$$NDCG_p = \frac{DCG_p}{IDCG}$$  \hspace{1cm} (9)

D. Performance

When evaluating our algorithm, we separate the evaluation into three parts. The first part is to compare the performance on video and text documents. The second part is to handle some multi-media documents which can be handled either by keeping their original ranking positions unchanged or by assigning the importance scores as their final ranking scores. We will discuss which one is better. The last part is to compare NDCG values when using different scores of \(\theta\) in our method.

As mentioned above, we crawl down the top 600 web pages returned by Google for each query. To make the re-ranking task more reasonable, we divide the pages into six groups, with 100 pages in a group. In this way, web pages originally in the first group will still keep in it after re-ranking. This will avoid the re-ranking positions are excessively distorted by noises.

1) Comparison result on video and text in our algorithm: Our method is based on semantic similarity which is inadequate to deal with video information. However, we put forward two specific methods to handle non-text documents. To test the adaptability of our method, we issue several video-concerned queries and several text queries. For example, we search the movie ‘Roman Holiday’, which may return a lot web pages containing video information. Meanwhile, we issue some queries, like cancer, which does not contain much video information. These queries may return a lot web pages containing texts. We process the documents with two methods mentioned above and show the results in Fig. 2.

In Fig. 2, the text line indicates the NDCG value of text type queries ranking results using our algorithm in a given proportion. The video line shows the same thing with video type queries words.

From Fig. 2, we can get the answer clearly that our algorithm is not very useful when searching some queries which will return web pages containing videos and few words. Because the semantic similarity is more focused on the relationship between the concepts of words, our method could not judge which video is better. For example, when we search a movie name, we may want to watch this movie, or get the music and comments about this movie. But we can only evaluate the text in web page other than video, the search result will not be good.

2) Remain unchanged or changed: Our method does not deal with multi-media data. But when we search queries, these data may be returned by search engine as results. We have two methods to deal with them: putting this document remain unchanged when sorting or giving them an importance to re-rank with others result. And we use the same evaluation to judge which one is better.

<table>
<thead>
<tr>
<th>(\theta)</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>unchanged</td>
<td>0.9201</td>
<td>0.9241</td>
<td>0.9273</td>
<td>0.9303</td>
<td>0.9315</td>
</tr>
<tr>
<td>changed</td>
<td>0.9007</td>
<td>0.9064</td>
<td>0.9101</td>
<td>0.9153</td>
<td>0.9209</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\theta)</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>unchanged</td>
<td>0.9392</td>
<td>0.9437</td>
<td>0.9494</td>
<td>0.9520</td>
<td>0.9492</td>
<td>0.9487</td>
</tr>
<tr>
<td>changed</td>
<td>0.9267</td>
<td>0.9318</td>
<td>0.9393</td>
<td>0.9429</td>
<td>0.9491</td>
<td>0.9487</td>
</tr>
</tbody>
</table>

In Table II, the ‘unchanged’ indicates the result page which only contains multimedia information just stays at its original position. The ‘changed’ indicates although the result page contains almost no text information, it will participate the re-ranking process with its importance score.

The data above seems that we prefer to keep its place unchanged, because giving this web page a value is not more reliable than using Google’s result information. For example, when we search philosophy, there are some useful and valuable web pages returned in PDF documents that we prefer remain unchanged. As shown in Fig. 2, the blue line’s NDCG value is always better than the red one. This phenomenon completely proves our point.

3) NDCG of different proportion in our method: We use remain unchanged strategy to do the experiment and in this experiment we almost search text queries that hardly have video search results.

In Fig. 3, we can observe that the best result is obtained when \(\theta=0.8\). When \(\theta=0\), it gets the worst result. When \(\theta=1\)
is less than 0.9, with the increasing of $\theta$, the result becomes better. We also find out that the result given by only using Google’s result is not as good as mixed with our algorithm.

Volunteers find something meaningful when evaluating. For example, when we search cancer, user may want to know what is cancer or the disease related with cancer. But Google returns a lot cancer foundations, which are not very suitable. They also find that there are some web pages which are dead links. Those are ranked far behind according to our algorithm.

As our method is based on calculating the distance between two concepts, our method can give better results when the users want to know the concept about query. But because of the diversity of words and subjectivity of volunteers, the final result sways in a certain extent. For example, when we issue 100 queries, the best result is obtained when $\theta=0.7$, while the best result is obtained when $\theta=0.8$ when we issue 200 queries. This rate is swaying between 0.7 and 0.9.

V. RELATED WORK

Researchers have done a lot of work on re-ranking. Jaroslaw Balinski and Czeslaw Daniłowicz propose a new method of document re-ranking that using inter-document relationships [1]. These relationships are expressed by distances and can be obtained from the text, hyperlinks or other information. The re-ranking method is based on a model of connections between documents and ensures that estimates of relevance weights are improved. The similar work focusing on re-ranking is reported by Lee et al [8]. They propose a model for an information retrieval system that is based on a document re-ranking method using clusters.

Apostol Natsev et al mainly focus on studying the problem of semantic concept-based query expansion and re-ranking for multimedia retrieval [11]. In particular, they explore the utility of a fixed lexicon of visual semantic concepts for automatic multimedia retrieval and re-ranking purposes. Chidlovskii et al propose system architecture for coupling user and community profiling to the information search process [2]. The search process and the ranking of relevant documents are accomplished within the context of a particular user or community point of view.

As an improvement, Lin et al propose a re-ranking method for reordering the images retrieved from an image search engine [9]. The ranking process is based on a relevance model, which is a probabilistic model that evaluates the relevance of the HTML document linking to the image, and assigns a probability of relevance. Also, some researchers do re-ranking search on video. Liu et al propose a PageRank-like graph-based approach to re-rank text-based search results [10]. To better exploit the underlying relationship between video shots, their proposed re-ranking scheme simultaneously leverages textual relevancy, semantic concept relevancy, and low-level-feature-based visual similarity. According to new research, Vidit Jain and Manik Varma [4] their objective is using the number of images clicked in response to a query to improve the performance of keyword based image search engines by re-ranking their original results.

VI. CONCLUSION

In this paper, we propose a method to improve the search quality by adding semantic similarity as a factor to the ranking result. We fetch the top N results returned by search engine, and use semantic similarities between the candidate and the query to re-rank the results. We first convert the ranking position to an importance score for each candidate. Then we combine the semantic similarity score with this initial importance score and finally we get the new ranks. In the experiment, we make 3 experiments and we use NDCG to evaluate the re-ranking results. According to experiment result, we find that our method doesn’t fit video search and when $\theta$ be set as 0.8 can get the best result. Also we find that the best proportion is swaying between 0.7 and 0.9. But whatever it says our algorithm makes sense. So we conclude that adding semantic information can enhance the search quality and meet users’ need to a certain extent.

REFERENCES