EVALUATING RERANKING METHODS USING WIKIPEDIA FEATURES

Koji Kurakado
Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka City, Fukuoka, Japan

Tetsuya Oishi†, Ryuzo Hasegawa‡, Hiroshi Fujita‡, Miyuki Koshimura‡
†*Research Institute for Information Technology, Kyushu University, Fukuoka, Japan*
‡*Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan*

Keywords: Wikipedia, Reranking, Link analysis.

Abstract: Many people these days access a vast document on the Web very often with the help of search engines such as Google. However, even if we use the search engine, it is often the case that we cannot find desired information easily. In this paper, we extract related words for the search query by analyzing link information and category structure. we aim to assist the user in retrieving web pages by reranking search results.

1 INTRODUCTION

In recent years, many people got possible to access Web very easily thanks to the vast spread of internet as well as the availability of convenient search engines. For instance, Google[1],Yahoo[2] are commonly used. Given a few keywords, these systems retrieve such Web pages that users want to see from among the huge databases residing on the internet. Google, in particular, successfully presents us the most suitable pages on the first page of the retrieval results by applying the PageRank algorithm (Page et al., 1998) which evaluates relevance of pages based on page links.

Nevertheless, since the Web sources are so enormous and constantly increasing, it is often the case that we are not satisfied with the results given by them. To solve the problem, we propose reranking methods based on Wikipedia. Wikipedia attracts attention on the field of NLP and Data-Mining, because of its impressive characteristics.

We implement a reranking system that extracts related words from a given search query. The system uses Wikipedia's link information and category structures.

Wikipedia is a Wiki-based huge Web encyclopedia. As a corpus for knowledge extraction, Wikipedia has several useful features. Thus, there have been various Wikipedia studies.

Semantic relatedness measurement is one of the most major Wikipedia studies. Strube and Ponzetto (Ponzetto and Strube, 2006) were the first to compute measures of semantic relatedness using Wikipedia. Their approach uses the category hierarchy of Wikipedia. Gabrilovich and Markovitch (Gabrilovich and Markovitch, 2007) proposed the Explicit Semantic Analysis (ESA) method. ESA represents the meaning of texts in a high-dimensional space of concepts derived from Wikipedia page. The semantic relatedness between two words is computed by the cosine similarity between the two vectors. They achieve the most accurate results using the gWordSimilarity-353 test collectionh (Finkelstein et al., 2002). Witten and Milne (Witten and Milne, 2008) proposed a new method based on link co-occurrence. Although the accuracy of this approach is a little worse than ESA, it requires far less data and resources. Chernov et al. (Chernov et al., 2006) extracted a category set by using links that direct to or refer to pages included in categories. According to their results, inlinks have superior performance in comparison to outlinks.

Researches that try to extract useful knowledge from Wikipedia are called "Wikipedia Mining" by Japanese researchers. Nakayama et al. (Nakayama et al., 2007) proposed a method, named pfibf, to com-

[1]http://www.Google.co.jp/
[2]http://www.Yahoo.co.jp/

Kurakado K., Oishi T., Hasegawa R., Fujita H. and Koshimura M..
EVALUATING RERANKING METHODS USING WIKIPEDIA FEATURES.
DOI: 10.5220/0003183003760381
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 376-381
ISBN: 978-989-8425-40-9

pute related words by analyzing the link structure of Wikipedia pages. They construct a huge scale association thesaurus. Ito et al. (Ito et al., 2008) proposed the method that constructs an association thesaurus, too. Their approach computes semantic relatedness by using link co-occurrence. They mention that the method is more scalable than pfibf in spite of their high accuracy close to that of pfibf. The method is similar to Milne's in that they use co-occurrence. Although the accuracy of this method is a little worse than Milne's, we consider the comparison should be made in the same environment. So, we cannot say that one is better than the other.

Nakayama et al. (Nakatani et al., 2009) proposed an evaluation method of search results by analyzing the link information and category structure of Wikipedia. They extract a category domain of query and evaluate search results by using terms included in the domain. Hori et al. (Hori et al., 2010) use Wikipedia as a source for query expansion.

2 RERANKING WEB SITES

In this section, we describe our reranking methods based on Wikipedia. We calculate the evaluation value for each site to which the search engine returns. Then we rerank the search result in descending order according to the evaluation value.

We select four features from Wikipedia for reranking: inlink, outlink, link co-occurrence, category. Here we propose a web page evaluation method that uses each of the above features.

An inlink of a Wikipedia page is a link pointing to that particular page. An outlink of a Wikipedia page is a link from that particular page to other page. For example, If page A contains a link directing to page B, A has an outlink to B, and B has an inlink from A.

In the following, we first show an approach that expands the category a search query belongs to. We consider not only the categories a query belongs to but also those related to the original categories as a category set for the query. Second, we explain the web page evaluation model that uses each of the features.

2.1 Expanding Categories to which a Query Belongs

A Wikipedia page belongs to one or more categories. In addition, unlike a thesaurus such as WordNet, the category structure of Wikipedia is not just hierarchical. It can be thought of as consisting of overlapping trees.

Nakayama et al. consider that they cannot get enough information from the category that the query originally belongs to. So, they regard the categories that contain a lot of inlinks to the query as those the query belongs to.

Suppose that c is a Wikipedia category and $size(c)$ is the total number of pages belonging to c. In addition, the number of pages in c from which the query has inlinks is expressed as in(c). $CScore_{in}(c)$ is defined as follows:

$$CScore_{in}(c) = \frac{in(c)}{size(c)} \quad (1)$$

We also use the technique that expands the category a query belongs to. Moreover, we propose the methods that uses outlink, link co-occurrence, or a category structure.

Expanding Categories based on Outlink

The method using outlink is very simple. We regard the categories that contain a lot of outlinks to the query as those the query belongs to. Thus, the score $CScore_{out}(c)$ is calculated as follows:

$$CScore_{out}(c) = out(c) \quad (2)$$

where $out(c)$ is the number of pages in c to which the query has outlinks.

Expanding Categories based on Link Co-occurrence

Thinking simply, link co-occurrence means that link A and link B appear in the same page. However, two links co-occur if they appear in a window of K sentences (window K in short).

Meanwhile, Wikipedia has the hierarchical paragraph from level2 to level4. For example, the Wikipedia page of "Computer" has level2 paragraph "Function". And level3 paragraphs "Control unit" and "Memory" belong to "Function".

Thus, we proposed three methods as follows:

1. Two links co-occur if they appear in the same page.
2. Two links co-occur if they appear in a window K.
3. Two links co-occur if they appear in the same paragraph whose level is highest and that contains more sentences than a window K.

Now, we show an example of the third method. Suppose that the size K of a window is 10 , the page of "Computer" contains 100 sentences , paragraph "Function" contains 15 sentences, paragraph "Memory" contains 5 sentences and that there is an inlink to "DRAM" in paragraph "Memory".

First, we look at paragraph "Memory" with the highest level. Then, since the number of sentences of paragraph "Memory" is less than that of a window K, we look at paragraph "Function", whose level is one higher than "Memory". Since the number of sentences of "Function" is larger than that of a window K, the links in "Function" co-occur with "DRAM".

The score $CScore_{co}(c)$ is calculated as follows:

$$CScore_{co}(c) = \frac{co(c)}{size(c)} \quad (3)$$

where $co(c)$ is the number of pages in c that co-occur with the target page of a query.

Expanding Categories based on Category Tree

On the category tree of Wikipedia, the categories in a near position have high relevance each other. Thus, we calculate the score $CScore_{cat}(c)$ as follows:

$$CScore_{cat}(c) = \frac{1}{2^{length(c)}} \quad (4)$$

where $length(c)$ is the number of paths from c to the category a target page of query belongs to. We calculate $length(c)$ only about the categories c_q the query belongs to, and parent categories of c_q, and children categories of c_q, and the categories that have common parents with c_q.

2.2 Web Site Evaluation based on Wikipedia Features

We evaluate web sites using the entries of Wikipedia included in the web sites. We compute a level expressing how much each entry of Wikipedia is related to the query according to the model that uses Wikipedia features. When a site includes a lot of highly related entries, we consider the site is important. The evaluation method based on the hypothesis that Wikipedia is a reliable corpus and the Wikipedia's entries closely related with query is important.

When a web site contains the entries of Wikipedia $w(s) = \{t_1, t_2, \ldots, t_n\}$, the score $SiteScore(s)$ is calculated as follows:

$$SiteScore(s) = \sum_{t \in w(s)} Score(t) \quad (5)$$

where $Score(t)$ is a related level of the Wikipedia's entry t with the query q. It is calculated according to the model described next.

Calculating a Score based on Inlink

The score calculated by using inlink $Score_{in}(p)$ is calculated as follows:

$$Score_{in}(p) = \frac{inlink(p)}{linknum(p)} \quad (6)$$

where $inlink(p)$ is the number of inlinks from a page of Wikipedia p to query q. $linknum(p)$ is the total number of links included in p.

Calculating Scores based on Outlink

To do this, we consider two cases as follows:

(1) based on TF-IDF

(2) based on the vector of TF-IDF

The score of (1), $Score_{outtfidf}(p)$, is calculated as follows:

$$Score_{outtfidf}(p) = \frac{outlink(p)}{linknum(q)} \cdot \log \frac{|W|}{|P|} \quad (7)$$

where $outlink(p)$ is the number of outlinks from query q to a page of Wikipedia p. $|W|$ is the total number of links in Wikipedia. $|P|$ is the document frequency of the entry of Wikipedia p.

Next, the method (2) is used by (Witten and Milne, 2008; Nakayama et al., 2007). Calculating TF-IDF in a page, they extract a vector of weighted links. After extracting the vectors for each page, relatedness between two pages can be calculated comparing their vectors by using cosine metrics. Thus, the score of (2), $Score_{outtfidfvec}(p)$, is calculated as follows:

$$Score_{outtfidfvec}(p) = \frac{\sum_{k=1}^{n} l_{pk} l_{qk}}{\sqrt{\sum_{k=1}^{n} l_{pk}^2} \sqrt{\sum_{k=1}^{n} l_{qk}^2}} \quad (8)$$

where $v_p = \{l_{p1}, l_{p2}, \ldots, l_{pn}\}$ is the vector of page p.

Calculating Scores based on Link Co-occurrence

To do this, we consider three cases as follows:

(1) using cosine metrics

(2) using the second-order co-occurrence (Schutze and Pedersen, 1997)

(3) based on Normalized Google Distance (Cilibrasi et al., 2007)

The score of (1), $cooOccur(p)$, is calculated as follows:

$$Score_{cocos}(c) = \frac{cooOccur(p)}{\sqrt{f(p) \cdot f(q)}} \quad (9)$$

where $f(p)$ is the term frequency of a page p. $cooOccur(p)$ is the number of pages that co-occur with the target page of a query.

Next, the method (2) is used by (Ito et al., 2008). They create a vector of link first-order co-occurrence. A first-order co-occurrence is calculated like $cooOccur(p)$ by cosine metrics . vi, which is the link vector of page p, is defined by the following formula: $v_p = \{c_{p1}, c_{p2}, \ldots, c_{pn}\}$ where c_{pi} is a first-order co-occurrence between page p and i. Thus, the score of (2) $Score_{cocosvec}(p)$ is calculated as follows:

$$Score_{cocosvec}(p) = \frac{\sum_{k=1}^{n} c_{pk}c_{qk}}{\sqrt{\sum_{k=1}^{n} c_{pk}^2}\sqrt{\sum_{k=1}^{n} c_{qk}^2}} \quad (10)$$

The method (3) is used by (Witten and Milne, 2008). They use Normalized Google Distance model. Thus, the score of (3) $Score_{congd}(p)$ is calculated as follows:

$$Score_{congd}(p) = \frac{\log(max(|P|,|Q|)) - \log(|P \cap Q|)}{\log(|W|) - \log(min(|P|,|Q|))} \quad (11)$$

where $|P \cap Q|$ is the number of a page including both the links of p and q.

Calculating Scores based on Category

To do this, we consider two cases as follows:

(1) using a set of categories to which the original query belongs

(2) using a set of categories that are expanded by the method described in section 3.1.

The score of (1), $Score_{cat}(p)$, is calculated as follows:

$$Score_{cat}(p) = \sum_{c \in C_{set}(q)} \frac{b(p,c)}{size(c)} \quad (12)$$

where $C_{set}(q) = \{c_1, c_2, \ldots, c_n\}$ is the set of categories to which a query originally belongs. $b(p,c)$ is a Boolean value. If a page p belongs to a category c, $b(p,c)$ becomes 1. Otherwise, $b(p,c)$ becomes 0.

Next, the score of (2), $Score_{catex}(p)$, is calculated as follows:

$$Score_{catex}(p) = \sum_{c \in C_{setex}(q)} \frac{b(p,c) \cdot CScore(c)}{size(c)} \quad (13)$$

where $C_{setex}(q)$ is the top K categories in descending order of $CScore(c)$.

3 EXPERIMENTS

In this section, we evaluate and compare our methods in terms of their performance in improving the search results for the initial query.

We used the Japanese Wikipedia database dump from 28 March 2010 in our experiment. We also use "Google Japanese search" in our experiment. We asked 6 evaluators to make 51 queries that contain at least one entry of Wikipedia and goals. They consist of 17 queries with at most three words for each. In addition, if the query contains a word which appears in disambiguation pages, we ask evaluators to select a concept. Next, we evaluated 100 web sites on a scale of 1 to 4 as follows: 4:"Highly relevant", 3:"Relevant", 2:"Partially relevant" and 1:"No relevant".

We remove web sites that do not contain more than 50 entries of Wikipedia from search results. Many sites that we remove are the html documents that we fail to parse. As a result, the average number of the search results become 93.86 sites. While the average number of sites evaluated as 3 or 4 is 18.02, the average number of sites evaluated as 4 is 6.34. Moreover, even if a site contains 500 or more entries of Wikipedia, we analyze up to 500 words in the site so that a long document does not have an advantage.

The accuracy of the results is measured by precision at K, and MAP (Mean Average Precision). Precision at K is precision of top K results. K is set to 10 in this experiment. MAP is an average of AP(Average Precision). AP is the average of ratios of the number of documents that user judges relevant to the number of whole given documents. Since these evaluation methods require that a documents is either relevant or irrelevant, we calculate both results for the strict relevance that we regard 4 as relevant and the relaxed relevance that we regard 3 and 4 as relevant.

4 RESULTS AND DISCUSSION

We have proposed several methods in section 3. In this section, we first evaluate category expansion methods. Second, we evaluate methods based on Wikipedia features. Finally, we compare our methods with Google search results. When we evaluate a combination of each method, we normalize each vector according to cosine normalization and add vectors.

Evaluation of Category Expansion Methods. Table 1 shows the results of category expansion methods in section 3.1. Where $P@10$ is the result for precision at K with the relaxed relevance (3 + 4). $P@10_H$ and MAP_H are the results for the strict relevance (4). The window size of link co-occurrence is set to 10. Here, we use the top 20 categories in descending order of score. Query in Table 1 is the method using a set of categories to which a query originally belongs.

In comparison with other methods that use only a single feature, outlink method is more accurate. In addition, the co-occur method using paragraph performed less accurately than the method using sen-

tence. Since we count all words in a paragraph that contains K and more sentences, there is the variation of the number of the co-occurrence links among pages. Moreover, the variation gives bad influence in the result.

It also shows that a combination of outlink and category is the best accurate method. Outlink achieves good results, while global information of category improves the outlink method. Since the method is more accurate than a method using a set of categories to which a query originally belongs , category expansion methods are effective.

Table 1: P@10 and MAP : category expansion methods.

	P10	P10_H	MAP	MAP_H
query	0.284	0.109	0.310	0.17
inlink	0.29	0.121	0.315	0.19
outlink	**0.304**	0.128	**0.325**	0.196
co-occur(all)	0.265	0.105	0.288	0.167
co-occur(sentence)	0.281	0.109	0.3	0.172
co-occur(paragraph)	0.273	0.104	0.296	0.167
outlink+category	**0.306**	0.122	**0.329**	0.197

Evaluation of the Methods based on Wikipedia Features. Table 2 shows the results of our methods using each Wikipedia feature in section 3.3. Where *count* is the average number of related words extracted from Wikipedia. Category in Table 2 corresponds to the method of expanding categories using outlink and category information. In comparison with other single methods, it identifies outlink as the more accurate measure, too. This result demonstrates that the complicated methods and methods of using a lot of information are bad. For example, the second order link co-occurrence method and the TF-IDF vector method are worse than simple methods. Thus it reveals that the number of related words does not reflect accuracy and deeply related words are very important.

Next, the result shows that a combination of outlink and category or link co-occurrence is the best accurate method. So, global information of category or link co-occurrence would improve the outlink method.

Comparison of our Methods with Google. Table 3 shows the results of Google and our methods. Our methods are considerably worse than Google search results. However, precision score of the search result is $18.02/93.86 = 0.192$. Thus, our method is effective in comparison with the case when we select sites at random.

Table 4 shows the results of Google and our methods when we use only one word query. The average number of the search results become 91.28 sites. While the average number of sites evaluated as 3 or more is 15.88, the average number of sites evaluated as 4 is 6.03.

Table 2: P@10 and MAP : the methods based on Wikipedia features.

	P@10	P@10_H	MAP	MAP_H	count
category	0.306	0.122	0.329	0.197	4433
inlink	0.301	0.127	0.321	0.208	804
outlink_1(tfidf)	**0.31**	0.129	**0.334**	0.222	143
outlink_2(tfidfVec)	0.301	0.126	0.318	0.2	83694
co-occur_1(cosine)	0.299	0.121	0.318	0.192	6271
co-occur_2(second)	0.267	0.108	0.283	0.159	166867
co-occur_3(NGD)	0.284	0.119	0.297	0.179	6271
outlink_1+category	0.318	0.133	**0.344**	0.224	4521
outlink_1+co-occur_1	**0.321**	0.136	0.342	0.225	6321

Table 3: P@10 and MAP : Google and our methods.

	P@10	P@10_H	MAP	MAP_H
outlink_1	0.31	0.129	0.334	0.222
outlink_1+category	0.318	0.133	0.344	0.224
outlink_1+co-occur_1	0.321	0.136	0.342	0.225
Google	0.494	0.239	0.505	0.417

Table 4: P@10 and MAP : Google and our methods when we use only one word query.

	P@10	P@10_H	MAP	MAP_H
category	**0.355**	0.15	0.373	0.262
outlink_1	0.347	0.157	**0.38**	0.314
outlink_1+category	0.365	0.167	**0.394**	0.311
outlink_1+co-occur_1	**0.382**	0.172	**0.394**	0.312
Google	0.439	0.231	0.477	0.474

Compared with the case where all queries are used, it is a very good result. The reason for this is that our methods have some difficulty to recognize deep semantic relationship between different words in a query unless the relationship is apparent or very strong. For example, given a query like “C++, Java”, our methods works well. But, given a query like “iPod, backup”, the methods do not work well.

On the other hand, precision score of the search result is $15.88/91.28 = 0.174$. Thus, our method is quite effective in comparison with the case when we select sites at random. But our method was worse than Google search result.

Figure 1 shows the graphs of the results. We concentrate on how much the accuracy of the retrieval results is improved compared to those obtained by an existing engine. First, we calculate AP for the results given by Google. Next, each query is classified into 10 classes according to the value of AP, first 0, second 0.1 or less, and so on, and finally 1.0 or less. Then, we calculate MAP for each class of queries and for each method being compared. Each MAP value of the point where AP is less than 0.2 represents the accuracy of each method when using the queries for

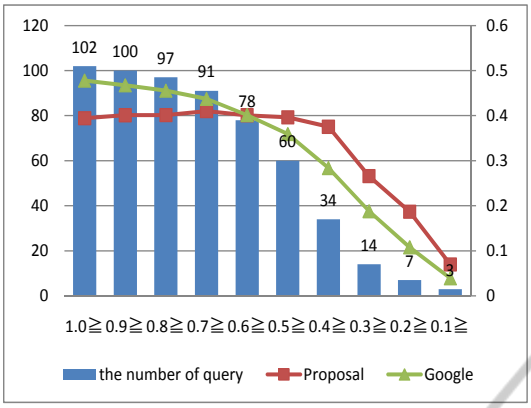

Figure 1: Reranking results.

which Google returns poor results. Each MAP value of the point that AP is less than 1.0 represents the accuracy of each method when using all queries.

The point where our method exceeds Google's AP is less than 0.6. This means our method is fairly effective when Google's result is not good.

5 CONCLUSIONS

We have shown that search results can be improved by reranking them with various methods based on Wikipedia features. Experimental results so far indicate the following.

- Category expansion methods are more effective than a method using a set of categories to which a query originally belongs.
- Reranking results are improved by deeply related words but not the number of related words.
- Basically simpler methods work better. However, more sophisticated methods, that are based on local weights of outlinks and inlinks, and global weights of link co-occurrence and category, work significantly well.
- Any Wikipedia feature works fairly well to improve search results.

Moreover, it turned out that outlinks are much better than inlinks to be used for weighting in our methods. This is interestingly quite contrary to the results by Chernov et al. When extracting statistical information from Wikipedia, we need to carefully choose an effective model. For this, we think a machine learning technique like Sumida et al. (Sumida et al., 2008) would be promising.

In the future research, we are going to extract more useful data by using Wikipedia features and classify data using the machine learning.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI (21500102).

REFERENCES

Chernov, S., Iofciu, T., Nejdl, W., and Zhou, X. (2006). Extracting semantic relationships between wikipedia categories. In *Proc. of Workshop on Semantic Wikis (SemWiki 2006)*. Citeseer.

Cilibrasi, R. et al. (2007). The google similarity distance. *IEEE Transactions on knowledge and data engineering*, pages 370–383.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin., E. (2002). WordSimilarity-353 Test Collection.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of IJCAI-07*, pages 6–12.

Hori, K., Oishi, T., Mine, T., Hasegawa, R., Fujita, H., and Koshimura., M. (2010). Related Word Extraction from Wikipedia for Web Retrieval Assistance. In *Proc. of ICAART 2010 vol.2*, pages 192–199.

Ito, M., Nakayama, K., Hara, T., and Nishio, S. (2008). Association thesaurus construction methods based on link co-occurrence analysis for wikipedia. In *Proc. of the 17th ACM conference on Information and knowledge management*, pages 817–826. ACM.

Nakatani, M., Jatowt, A., Ohshima, H., and Tanaka, K. (2009). Quality evaluation of search results by typicality and speciality of terms extracted from wikipedia. In *Database Systems for Advanced Applications*, pages 570–584. Springer Berlin/Heidelberg.

Nakayama, K., Hara, T., and Nishio, S. (2007). Wikipedia mining for an association web thesaurus construction. *Web Information Systems Engineering–WISE 2007*, pages 322–334.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web.

Ponzetto, S. and Strube, M. (2006). Wikirelate! computing semantic relatedness using wikipedia. In *Proc. AAAI-06*, pages 1419–1424.

Schutze, H. and Pedersen, J. (1997). A cooccurrence-based thesaurus and two applications to information retrieval. *Information Processing & Management*, 33(3):307–318.

Sumida, A., Yoshinaga, N., and Torisawa, K. (2008). Boosting precision and recall of hyponymy relation acquisition from hierarchical layouts in wikipedia. *In Proc. of the LREC 2008*.

Witten, I. and Milne, D. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proc. of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30.