EVALUATING RERANKING METHODS BASED ON LINK CO-OCCURRENCE AND CATEGORY IN WIKIPEDIA

Yuichi Takiguchi[1], Koji Kurakado[1], Tetsuya Oishi[2],
Miyuki Koshimura[2], Hiroshi Fujita[2] and Ryuzo Hasegawa[2]
[1]*Graduate School of Information Science and Electrical, Kyushu University, Fukuoka, Japan*
[2]*Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan*

Keywords: Wikipedia, Data-mining, Link analysis.

Abstract: We often use search engines in order to find appropriate documents on the Web. However, it is often the case that we cannot find desired information easily by giving a single query. In this paper, we present a method to extract related words for the query by using the various features of Wikipedia and rank learning. We aim at developing a system to assist the user in retrieving Web pages by reranking search results.

1 INTRODUCTION

Calculating relatedness measurement is a subject in natural language processing (NLP) and has been studied by many researchers. It gives a semantic relatedness between two words. For example, the relatedness between *computer* and *memory* is 0.9, while one between *computer* and *tomato* is 0.1. The measurement is a basic metric in several applications in data mining and mainly used to extract some related words of a word.

Wikipedia, which is a Wiki-based huge Web encyclopedia, attracts many researchers in NLP or data mining because of its impressive characteristics. We all can create new articles and edit existing ones in Wikipedia. It contains a wide range of diverse articles and new information since there are a huge number of editors.

There are many studies on calculating semantic relatedness measurement with Wikipedia. However, most of these studies do not seem to extract enough information from Wikipeida. There also exist many studies using two or more features of Wikipedia. Such studies calculate measurements by linearly summing or multiplying values derived from corresponding features. These measurements do not seem to be good metrics.

This paper proposes a method to calculate semantic relatedness by using rank learning. It uses several features of Wikipedia. These features are "Linking Structure", "Category", "Author's information" and so on. We expect that we can extract related words close to human sense with the method. We compare the method and the previous ones in the literature. Experimental results show usefulness of the method.

2 RELATED WORK

Strube and Ponzetto (Strube and Ponzetto, 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.

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.

Nakayama et al. (Nakayama et al., 2007) proposed a method to construct a large scale association thesaurus, by analyzing the link structure of Wikipedia with the PF-IBF model, that is based on TF-IDF. PF is calculated by considering the number of links from a particular page to other pages and the

Takiguchi Y., Kurakado K., Oishi T., Koshimura M., Fujita H. and Hasegawa R..
EVALUATING RERANKING METHODS BASED ON LINK CO-OCCURRENCE AND CATEGORY IN WIKIPEDIA.
DOI: 10.5220/0003729302770282
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 277-282
ISBN: 978-989-8425-95-9

distance between links, while IBF is calculated with the number of site pages that link to the particular page.

Ito et al. (Ito et al., 2008) also proposed the method that constructs an association thesaurus. Their approach computes semantic relatedness by using link co-occurrence. They mention that the method has precision as accurate as PF-IBF and requires less complexity. 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 cannot say that one is better than the other, since experimental environments are not the same.

Nakayama et al. (Nakayama 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.

The works mentioned about do not have good accuracy when an article has few links to other articles. Our goal is to get a good result from every article. In our previous work (Kurakado et al., 2011), we extracted related words by using the features of Wikipedia, and showed that search results can be improved by reranking them with various methods based on Wikipedia features. Our current research is along the same line as our previous work, yet is different in that the current method uses more features and rank learning.

3 EXTRACT FEATURES BASED ON WIKIPEDIA

We take the various features of Wikipedia, such as link structure and category structuref. These features are available for extracting semantic relatedness. Based on existing studies on the relatedness calculation, we extract some features for rank learning.

In the following subsections, we will explain each feature in detail.

3.1 Outlink

An outlink of a Wikipedia page is a link from that particular page to other page. To calculate the score of the feature in terms of outlink, we consider three methods. Here, we call an article explaining a word x "an article x" for short.

The first method, $F_{outlink1}(p)$, is defined as follows:

$$F_{outlink1}(p) = \frac{LF(p,key)}{\sum_{x \in W} LF(x,key)} \quad (1)$$

where *key* is a Wikipedia entry from which we try to extract related words, and p is other Wikipedia entry except *key*, that is the entry of a related word candidate. W represents all articles on Wikipedia and $LF(x,y)$ is the number of occurrences of link x in an article y.

The second method applies TF-IDF to links. The score, $F_{outlink2}(p)$, is defined as follows:

$$F_{outlink2}(p) = \frac{LF(p,key)}{\sum_{x \in W} LF(x,key)} \cdot \log \frac{|W|}{|P|} \quad (2)$$

where $|W|$ is the total number of articles in Wikipedia, and $|P|$ is the document frequency of the entry of a Wikipedia article p.

The score with the third method, $F_{outlink3}(p)$, is defined as follows:

$$F_{outlink3}(p) = \frac{\sum_{i=1}^{n} l_{p_i} l_{key_i}}{\sqrt{\sum_{i=1}^{n} l_{p_i}^2} \sqrt{\sum_{i=1}^{n} l_{key_i}^2}} \quad (3)$$

where $v_p = \{l_{p1}, l_{p2}, \ldots, l_{pn}\}$ is the TF-IDF vector of article p.

3.2 Inlink

An inlink of a Wikipedia page is a link pointing to that particular page. This is just opposite of outlink. We consider the following methods to calculate the score of the feature in terms of inlink.

The first method uses the frequency of occurrence of links from article p to article *key*, and The score with the first method is defined as follows:

$$F_{inlink1}(p) = \frac{LF(key,p)}{\sum_{x \in W} LF(x,p)} \quad (4)$$

The second method applies the TF-IDF vector used for the outlink feature to the frequency of inlink occurrences.

3.3 Link Co-occurrence

Link co-occurrence is a method that applies the idea of word co-occurrence to inlinks for a Wikipedia article. As stated in (Ito et al., 2008), we consider that words co-occur if they appear within a certain distance.

Figure 1 shows an example when the window size is three. Underlined characters represent inlinks of Wikipedia. In this figure, B co-occurs with A,C,D,and E. In the actual experiments, the window size is set to 10.

To calculate the score of the feature in terms of link co-occurrence, we consider a method that uses

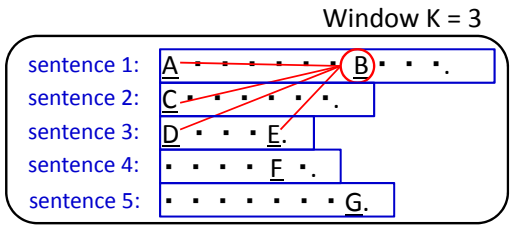

Figure 1: An example of link co-occurrence.

cosine metrics.The score , $F_{coOccur1}(p)$, is defined as follows:

$$F_{coOccur1}(p) = \frac{coOccur(key, p)}{\sqrt{f(key) \cdot f(p)}} \quad (5)$$

where $f(p)$ is the number of occurrences of an article p in all of the Wikipedia articles, $\sum_{x \in W} LF(p,x)$, and $coOccur(p,q)$ is the number of links that co-occurr between articles p and q.

3.4 Category Structure

3.4.1 Expanding Categories

The categories of Wikipedia form a tree strucure. The categories in a near position have high relevance each other. So, for the category c_{key} the *key* article belongs to, we set a relevance score to parent categories of c_{key}, children categories of c_{key}, and the categories that have common parents with c_{key}. The relevance score given to category c, $CategoryScore1(c)$, is defined as follows:

$$CategoryScore1(c) = \frac{1}{2^{length(c)}} \quad (6)$$

where $length(c)$ is the number of paths from c to the category a target article belongs to. Thus, the score of category c, $CategoryScore2(c)$, is defined as follows:

$$CategoryScore2(c) = \frac{out(c)}{\log(size(c))} \quad (7)$$

where $out(c)$ is the number of outlinks from *key* to the articles that belong to category c. Finally, after performing morphological analysis for both the title of *key* and categories extracted from the category tree, we take the agreement degree of nouns of *key* and each category as a relevance score. Moreover, we normalize the relevance scores such that the maximum value for each is one, and take the summation of the normalized scores as a relevance score. Top 10 highest scoring categories of the extracted one are classified to the expanded categories.

3.4.2 The Feature using the Expanded Category

Let C_{ex} be a set of expanded categories. Then the relevance score of each category, $F_{cateogry_2}(p)$, is calculated as follows:

$$F_{cateogry_2}(p) = \sum_{c \in C_{ex}} \frac{b(p,c) \cdot CategoryScore(c)}{size(c)} \quad (8)$$

3.5 Other Features

In addition to the above features, we adopt the following features.

3.5.1 Links in Definition Sentences

A lot of articles in Wikipedia have sentences explaining its concept in the beginning. They are called definition sentences. An example is shown in Figure 2. We define the scores of the features in terms of outlink and inlink by using only the link information in the definition sentences. These scores are obtained by the same methods in Section 3.1 and 3.2.

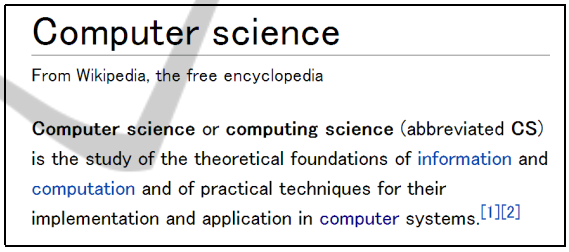

Figure 2: An example of a definition sentence.

3.5.2 Degree of Agreement among Morphemes of Article Names

We consider that articles containing common words in their name are deeply related to each other. So, we adopt the degree of agreement among morphemes of article names as a feature. When two articles x and y are given, their relatedness is calculated as follows:

1. We perform morphological analysis of x and y, and extract morphemes of the noun. Here, we assume $M_x = \{m_{x1}, m_{x2}, \ldots, m_{xn}\}$ is a morpheme vector of x.
2. For each element of M_x and M_y, we check whether one morpheme is a prefix of the other morpheme. If there is such a pair of morphemes, we set the relatedness between articles x and y, $morpSim(x,y)$ to 1. Otherwise, we set it to 0.

Therefore, the score of the feature using the agreement degree of morphemes of article names is defined as follows:

$$F_{morpSim}(p) = morpSim(key, p) \quad (9)$$

3.5.3 Related Item

Articles in Wikipedia have section headers such as "outline", "external link", and "related item". An example of related item is shown in Figure 3. Taking into account links in the related item, we define the score of the feature using links appearing in the related item of an article as follows.

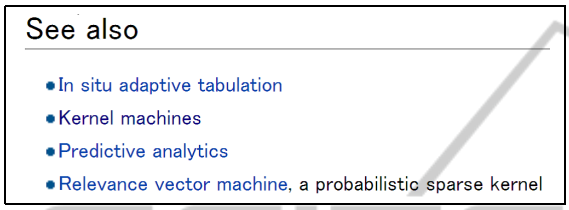

Figure 3: An example of a related item.

$$F_{relate}(p) = relate(key, p) + relate(p, key) \quad (10)$$

where $relate(x)$ is the number of occurrences of a link x in the related item of an article y.

3.6 Extracting Features based on a Search Engine

To calculate relatedness, several methods have been proposed that use the number of hits for a query in the search engine. They include WebPMI and Google Distance (NGD). NGD is given by the following formula:

$$NGD(x,y) = \frac{\max(\log f(x), \log f(y)) - \log f(x,y)}{\log M - min(\log f(x), \log f(y))} \quad (11)$$

Where M is the total number of indexes of a search engine, $f(x)$ is the number of results retrieved by a query x, and $f(x,y)$ is the number of results retrieved by queries x and y, The larger NGD is, the smaller the relatedness is. Thus, we use the score defined below:

$$F_{NGD}(p) = 1.0 - NGD(key, p) \quad (12)$$

The methods based on the number of hits, such as NGD, work well when two words are indexed in the same extent by the search engine.

On the other hand, it is often the case that words in Wikipedia appear in only a few documents that the search engine indexes. To deal with such words, we propose a method to calculate the relatedness and the score using it as follows:

$$webHit(x,y) = \frac{\min(\log f(x), \log f(y)) - \log f(x,y)}{min(\log f(x), \log f(y))} \quad (13)$$

$$F_{webHit}(p) = 1.0 - webHit(key, p) \quad (14)$$

4 EXPERIMENTS

We calculate semantic relatedness between two words with the Japanese Wikipedia. We exclude some articles unsuitable for the calculation. Such articles are ones with unsourced statements, ones for disambiguation, ones describing the year, and so on. We use the feature extraction based on Japanese Google search engine.

We extract candidates of related words and features, and calculate relatedness between words by using methods described in the previous section. We have six examiners evaluate relatedness for 2550 pairs of words. We regard the average evaluation of the six examiners correct relatedness in our experiments.

We select 2550 pairs of words as follows. For each article, we extract top 100 words related to the article. We select top 30 words from the 100 words and 21 words randomly from the remaining 70 words. We make 51 pairs of words such that one is the article name i.e. the key, and another is from the 51 words. These 51 pairs are shuffled and given to the six persons. Totally, they are given 2550 (50 keys $\times$ 51) pairs of words. We use P@K (Precision at K), MAP (Mean Average Precision), and NDCG (Normalized Discounted Cumulative Gain) as evaluation measurements for rank learning. In order to show usefulness of the presented method, we compare the following methods.

- Base Line : the linear sum of features described in the previous section.
- SVM regression, Ranking SVM, SVM-MAP, RankNet, RankBoost, AdaRank, and Coordinate Ascent.
- Relatedness based on Wikipedia thesaurus (Nakayama et al., 2009).

We use a public WebAPI as a Wikipedia thesaurus. We make the learning machine based on listwise approach so as to optimize NDCG@10. We use evaluations by 10-fold cross validation as precisions of all the above methods except base line and Wikipedia thesaurus, because these methods perform learning.

4.1 Results

Table 1 shows experimental results of the proposed methods. We regard the evaluations greater than or equal to 6 as positive for calculating MAP and P@10. We obtain the evaluation value, i.e., precision of each method by performing cross-validation on a data set of 2550 pairs of words used as training data. The cost parameter of the SVM regression is set to 200 and that of the ranking SVM is set to 150. In our experiment,

however, the parameter dose not influence the results so much. We use a linear kernel function for SVM. All features for a key are normalized so that their max value equals 1.

In addition to the features explained in Section three, we also use the squares of $outlink_2$, $category_2$, relate, $define_2$, and webHit. Thus, we use 19 features.

Table 1 tells us that methods using SVM are superior to others in all evaluation measurements. RankBoost and Coordinate Ascent are relatively good according to NDCG@10. RankNet and AdaRank are worse than Base Line.

Table 1: Results of the proposed method.

method	MAP	NDCG@10	P@10
Base Line	0.672	0.803	0.624
RankNet	0.672	0.800	0.624
AdaRank	0.680	0.805	0.634
SVM-MAP	0.723	0.830	0.682
Coordinate Ascent	0.740	0.838	0.682
SVM Regression	0.744	0.844	0.689
Ranking SVM	0.744	0.843	0.692
Rank Boost	0.746	0.844	0.696

Table 2 shows comparison of Wikipedia thesaurus and Ranking SVM. We get 300 words related to a word with a WebAPI of Wikipedia thesaurus. On the other hand, we get 30 words related to the word from the Web page of Wikipedia thesaurus. Accordingly, we regards both the 30 words from the Web page and 300 words from WebAPI as the related words from Wikipedia thesaurus.

There are several keys for which we cannot obtain related words from Wikipedia thesaurus. Additionally, there are many words which Ranking SVM evaluates but Wikipedia thesaurus does not mention. Therefore, for the comparison, we utilize words which both Wikipedia thesaurus and Ranking SVM deal with. Table 2 tells us that Ranking SVM is superior to Wikipedia thesaurus. This is a debatable point because we exclude many words for the comparison. According to the literature (Nakayama et al., 2009), Wikipedia thesaurus also utilizes a machine learning technique with training data different from ours. Thus, there is a room for further investigation.

Table 2: Comparison of Wikipedia thesaurus.

method	MAP	NDCG@10	P@10
Wikipedia thesaurus	0.670	0.820	0.500
Ranking SVM	0.761	0.853	0.561

4.2 Effect of Features

Table 3 shows the difference between the effect of an individual feature and that of the collection of features. All values are 10-fold of their original values. The last row (ALL) shows the evaluations with all features described in 3. The other rows show the evaluations with all features except the specific feature or the collection of features indicated in the first column. The number in parentheses indicates the difference between the evaluation and that of ALL. The outlinks collection is a collection of $outlink_1$, $outlink_2$, $outlink_3$, and $define_1$. The inlinks collection is a collection of $inlink_1$ and $define_2$. The search collection is a collection of ngd and webHit.

The table tells us that $inlink_1$ is the most effective feature and $define_2$ is the second most effective feature. Then, webHit, relate, $outlink_3$, $category_2$ and morpSim follow. The outlinks collection is the most effective collection of features because its sum of differences is the lowest. The inlinks collection and search collection are also effective in general because their sums of differences are small.

We tried several normalizations. Table 4 shows the evaluations of the five normalizations. These evaluations are obtained by Ranking SVM. The table tells us that the normalization, such that the maximum of each feature is 1 for each key, is the best one. This suggests that rank learning with key and a set of its related words is suitable for a task to extract good related words.

5 CONCLUSIONS

In this study, we extracted and scored various features from Wikipedia pages. We have proposed a method for extracting some related words by rank learning. As resources, not only Wikipedia but also information given by a search engine are used. Our proposed methods are able to find the suitable combination of features based on machine learning. The results indicate that Ranking SVM with combining various features achieves the best accuracy. Normalization experiments show that the framework of rank learning is effective for extracting related words. Compared to the Base Line and Wikipedia thesaurus, the best combination of learning machines contributes to improve accuracy more significantly.

In the future research, we are going to extract relatedness between two words and semantic relationship from Web by using machine learning, a probabilistic model and Web ontology.

Table 3: Effect of each feature.

	Rank Correlation	MAP	NDCG@10	P@10	total decrease
outlink1	55.4(0.0)	73.8(0.0)	84.1(-0.2)	70.5(-0.1)	-0.3
outlink2	55.2(-0.2)	73.1(-0.7)	84.3(0.0)	70.3(-0.3)	-1.2
outlink3	54.4(**-1.0**)	73.0(-0.8)	83.9(-0.4)	69.8(-0.8)	**-3.0**
inlink1	52.6(**-2.8**)	72.1(**-1.7**)	83.0(**-1.3**)	69.6(**-1.0**)	**-6.8**
inlink2	55.2(-0.2)	73.4(-0.4)	84.2(-0.1)	70.9(+0.3)	-0.4
co-occur1	55.2(-0.2)	73.6(-0.2)	84.1(-0.2)	70.5(-0.1)	-0.7
category2	54.5(-0.9)	73.4(-0.4)	84.0(-0.3)	70.1(-0.5)	**-2.1**
define1	55.3(-0.1)	73.5(-0.3)	83.9(-0.4)	70.1(-0.5)	-1.3
define2	54.0(**-1.4**)	72.7(**-1.1**)	83.4(-0.9)	68.5(**-2.1**)	**-5.5**
morpSim	55.1(-0.3)	73.1(-0.7)	83.8(-0.5)	70.2(-0.4)	-1.9
relate	54.5(-0.9)	72.7(**-1.1**)	83.9(-0.4)	69.8(-0.8)	**-3.2**
ngd	55.3(-0.1)	73.8(0.0)	84.1(-0.2)	70.6(0.0)	-0.3
webHit	54.4(**-1.0**)	72.6(**-1.2**)	84.0(-0.3)	69.7(-0.9)	**-3.4**
outlinks collection	48.6(-6.8)	69.7(-4.1)	80.7(-3.6)	65.8(-4.8)	**-19.3**
inlinks collection	48.3(-7.1)	69.6(-4.2)	82.2(-2.1)	66.5(-4.1)	**-17.5**
search collection	53.0(-2.4)	71.9(-1.9)	83.1(-1.2)	68.1(-2.5)	**-8.0**
ALL	55.4	73.8	84.3	70.6	-

Table 4: Comparison of normalization methods.

measures	Rank correlation	MAP	NDCG@10	P@10
Maximum 1 for each *key*	0.554	0.738	0.843	0.706
Mean square for each *key*	0.536	0.718	0.831	0.680
Maximum 1 in total	0.509	0.709	0.822	0.680
Mean square in total	0.504	0.694	0.810	0.656

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.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 6–12.

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

Kurakado, K., Oishi, T., Hasegawa, R., Fujita, H., and Koshimura, M. (2011). Evaluating Reranking Methods Using Wikipedia Features. In *Proc. of ICAART 2011*, pages 376–381.

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.

Nakayama, K., Ito, M., Hara, T., and Nishio, S. (2009). Wikipedia Relatedness Measurement Methods and Influential Features. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*, pages 738–743. IEEE.

Strube, M. and Ponzetto, S. (2006). WikiRelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1419. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

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