A web based method for Measuring Semantic Relatedness between words

Bo You
Department of Computer Science
and Technology
Central China Normal University
Wuhan, China
boyou0909@gmail.com

Tingting He
Department of Computer Science
and Technology
Central China Normal University
Wuhan, China
tthe@mail.ccnu.edu.cn

Fang Li
Department of Computer Science
and Technology
Central China Normal University
Wuhan, China
fang__lf@163.com

***Abstract*—Semantic relatedness measures play important roles in many fields, such as information retrieval and Nature Language Processing. There are mainly two kinds of traditional methods to measure semantic relatedness: dictionary based and corpus based. However, with the development of information technology, web search engine is used to do this work. In this paper, we propose a method integrating page counts and web-based kernel function for measuring semantic relatedness between words. It gets a better result than using page counts and web-based kernel function alone. Experimental results show Spearman rank correlation coefficient can reach 0.63 and Correlation reach 0.724.**

Keywords- Semantic relatedness, Web mining, Web search, Kernel functions

I. INTRODUCTION

The study of semantic relatedness between words or short text snippets is very important in information retrieval and natural language processing. Semantic relatedness is different from semantic similarity. For example, Microsoft is always associated with Bill Gates, we said Microsoft and Bill Gates are semantically related; mobile phone and cell phone have the same meaning, so the semantic similarity of mobile phone and cell phone is high. However, Semantic similarity and semantic relatedness are positive correlation with each other, that is, if the semantic similarity of two words is high, the two words must semantically relate to each other.

Generally speaking，there are two kinds of traditional methods computing the semantic relatedness of two words: one is dictionary based, for instance WordNet[1] or HowNet[2]; the other is corpus based, such as Wikipedia[3,4]. With the development of the information society, the search engine is become indispensable when people want to get network information. So we can try to use the web search engine to compute the semantic relatedness. Directly applying document similarity measures, such as the widely used cosine coefficient, it maybe gets irrelevant result. Bill Gates is the founder of Microsoft, but applying the cosine would yield a similarity of 0 since Bill Gates and Microsoft contain no common terms; however, in cases where two snippets may share terms, they may be using the term in different contexts. Consider the words "apple pie" and "Apple Computer", the former apple means a kind of fruit whereas the latter refers to a computer produced by Apple Company. Thus, while the cosine score between these two snippets would be 0.5 due to the shared lexical term "apple". The use of this shared term is not truly an indication of relatedness between words.

We propose a method which uses both page counts and search results. First of all, we treat each word as a query to a web search engine and find a number of documents which contain the terms in the original snippets, we use these returned documents to create a context vector for the original snippet, where such a context vector contains many words that tend to occur in context with the original snippet terms. Then, we use page counts measure the semantic relatedness of two words. At last, we use linear weighted sum to get the final result.

We introduce some methods which measure the semantic relatedness or similarity between words in Section 2. Computing semantic relatedness can be based on the result of semantic similarity. We then formally present our new method in Section 3. This is followed by experiment and result in Section 4. Finally, in Section 5 we provide some conclusions and directions for the future work.

II. RELATED WORK

There are two kinds of methods to measure semantic relatedness between words, dictionary based and corpus based.

WordNet is a lexical database for the English language. Budanitsky and Hirst (2006) [1] provide a survey of many WordNet-based measures of lexical similarity based on paths in the hypernym taxonomy. Among all this methods, an information-content–based measure proposed by Jiang and Conrath is found superior to others.

Michael Strube and Simone Paolo Ponzetto [2] using Wikipedia for computing semantic relatedness and compare it to Wordnet on various benchmarking datasets. They find integrating Google, WordNet and Wikipedia based measures get the best results on the largest available dataset. Evgeniy Gabrilovich and Shaul Markovitch[3] propose Explicit Semantic Analysis(ESA), a novel method that represents the meaning of texts in a high-dimensional space of concepts derived from Wikipedia. This method improves the relatedness score and makes it easy to human users.

Xuyun and Fanxiaozhong[4] computer semantic relatedness based on Hownet. The method can computer semantic relatedness using the resources of Hownet, and it get a satisfactory result.

Published by Atlantis Press, Paris, France.

With the development of the information society, some people try to computer semantic relatedness based on web search engine. Sahami et al., [5] measured semantic similarity between two queries using snippets returned for those queries by a web search engine. They collect snippets of two queries from a search engine and represent each snippet as a TF-IDF weighted term vector, then L_2 normalize vectors and computer the centroid of the set of vectors, at last, the inner product between the corresponding centroid vectors is the result of semantic similarity between two queries.

Danushka Bollegala et al., [6] also use web search engine to measure semantic similarity between words. They obtain page counts and snippets of each word from web search engine, using automatically extracted lexico-syntactic patterns from text snippets to computer semantic similarity, then integrate different similarity scores with support vector machine.

III. Method

A. Web-based Kernel Function Measuring Semantic Similarity

We use a new similarity function raised by Sahami et al.[5], and make some changes because it measures semantic similarity of Chinese words. Let *x* represents a short test snippet.

1. Issue *x* as a query to a search engine S.

2. Let *R(x)* be the set of (at most) *n* retrieved documents $d_1;d_2;\dots;d_n$

3. Compute the TFIDF term vector v_i for each document $d_i \in R(x)$

4. Truncate each vector v_i to include its *m* highest weighted terms

5. Let *C(x)* be the centroid of the L_2 normalized vectors v_i:

$$C(x)=\frac{1}{n}\sum_{i=1}^{n}\frac{v_i}{\|v_i\|_2} \tag{1}$$

6. Let *QE(x)* be the L_2 normalization of the centroid *C(x)*:

$$QE(x)=\frac{C(x)}{\|C(x)\|_2} \tag{2}$$

The goal is to measure semantic relatedness of words and phrases in Chinese, so we will alter some steps. When getting *n* retrieved documents $d_1;d_2;\dots;d_n$,we segment words for each document and remove stop words. In step 3, we consider a TFIDF vector weighting scheme [7], where the weight $w_{i,j}$ is defined to be:

$$w_{i,j}=tf_{i,j}\times\log(\frac{N}{df_i}) \tag{3}$$

Where, $w_{i,j}$ means the weight of term t_i in document d_j . $tf_{i,j}$ is the frequency of in d_j . *N* is the total number of documents in the corpus, and df_j is the total number of documents that contain t_i . Certainly, there are lots of other weighting schemes, but we use TFIDF because it performs better in this method.

When we use search engine to query, there are some query results that we do not need. In step 2, we use parts of retrieved documents rather than the entirety of retrieved documents to produce vectors. Most web search engines generate contextually descriptive text snippet for each document, so we choose the snippet to create vector. This will make our algorithm more efficient and get a more accurate result. Empirically, we found that using 10 retrieved documents can obtain a good result. Also, in step 4, we set the maximum number of terms in each vector *m* = 10, because we have found this value will get representational robustness, at the same time, it has good efficiency.

Finally, we define the semantic kernel function *K* as the inner product of the query expansions for two text snippets. There are two short text snippets: *x* and *y*, we define the semantic similarity kernel between them as:

$$K(x,y)=QE(x)\cdot QE(y) \tag{4}$$

Using the semantic kernel function *K*, we can get results which between 0 and 1, the greater the *K*, the more similar *x* and *y* are.

B. Page Counts based Method

Page count of a query is the number of pages that contain the query words. It is different from word frequency because the query words may appear many times in one page. We define page counts for the query *P* and *Q* as an approximation of co-occurrence of two words *P* and *Q* on the web.

However, we cannot measure semantic relatedness of two words *P* and *Q* with page counts for *P* and *Q* alone. For example, Google returns 605,000,000 as the page counts for “car” AND “bus”, whereas the same is 2,200,000,000 for “car” AND “phone”. Although, “car” is more semantic relatedness to “bus” than “phone”, page counts for query “car” AND “phone” are more than triple greater than those for the query “car” AND “bus”. So we should take page counts for the individual words *P* and *Q* into consideration.

We have used four popular co-occurrence measures; Jaccard, Overlap (Simpson), Dice, and PMI (Point-wise mutual information), to measure semantic relatedness using page counts.

The Jaccard coefficient for a pair of words (*P* and *Q*) is defined as:

$$WebJaccard(P,Q)=\begin{cases}0 & if\ H(P\cap Q)\le c\\ \dfrac{H(P\cap Q)}{H(P)+H(Q)-H(P\cap Q)} & otherwise.\end{cases} \tag{5}$$

The Overlap coefficient is defined as:

$$WebOverlap(P,Q)=\begin{cases}0 & if\ H(P\cap Q)\le c\\ \dfrac{H(P\cap Q)}{\min(H(P),H(Q))} & otherwise.\end{cases} \tag{6}$$

TABLE I. SEMANTIC RELATEDNESS OF HUMAN RATINGS AND BASELINES ON MILLER-CHARLES' DATASET

Word Pair	**Miller-Charles'**	**Web Jaccard**	**Web Dice**	**Web Overlap**	**Web PMI**	**Sahami [5]**	**Proposed method(α =0.6)**
cord-smile	0.13	0.102	0.108	0.036	0.207	0.090	0.1368
rooster-voyage	0.08	0.011	0.012	0.021	0.228	0.197	0.2094
noon-string	0.08	0.126	0.133	0.060	0.101	0.082	0.0896
glass-magician	0.11	0.117	0.124	0.408	0.598	0.143	0.325
monk-slave	0.55	0.181	0.191	0.067	0.610	0.095	0.301
coast-forest	0.42	0.862	0.870	0.310	0.417	0.248	0.3156
monk-oracle	1.1	0.016	0.017	0.023	0	0.045	0.027
lad-wizard	0.42	0.072	0.077	0.070	0.426	0.149	0.2598
forest-graveyard	0.84	0.068	0.072	0.246	0.494	0	0.1976
food-rooster	0.89	0.012	0.013	0.425	0.207	0.075	0.1278
coast-hill	0.87	0.963	0.965	0.279	0.350	0.293	0.3158
car-journey	1.16	0.444	0.460	0.378	0.204	0.189	0.195
crane-implement	1.68	0.071	0.076	0.119	0.193	0.152	0.1684
brother-lad	1.66	0.189	0.199	0.369	0.644	0.236	0.3992
bird-crane	2.97	0.235	0.247	0.226	0.515	0.223	0.3398
bird-cock	3.05	0.153	0.162	0.162	0.428	0.058	0.206
food-fruit	3.08	0.753	0.765	1	0.448	0.181	0.2878
brother-monk	2.82	0.261	0.274	0.340	0.622	0.267	0.409
asylum-madhouse	3.61	0.024	0.025	0.102	0.813	0.212	0.4524
furnace-stove	3.11	0.401	0.417	0.118	1	0.310	0.586
magician-wizard	3.5	0.295	0.309	0.383	0.863	0.233	0.485
journey-voyage	3.84	0.415	0.431	0.182	0.467	0.524	0.5012
coast-shore	3.7	0.786	0.796	0.521	0.561	0.381	0.453
implement-tool	2.95	1	1	0.517	0.296	0.419	0.3698
boy-lad	3.76	0.186	0.196	0.601	0.631	0.471	0.535
automobile-car	3.92	0.654	0.668	0.834	0.427	1	0.7708
midday-noon	3.42	0.106	0.112	0.135	0.586	0.289	0.4078
gem-jewel	3.84	0.295	0.309	0.094	0.687	0.211	0.4014
Correlation	1	0.259	0.267	0.382	0.548	0.579	0.724

The Dice coefficient is defined as:

$$WebDice(P,Q)=\begin{cases}0 & if\ \ H(P\cap Q)\le c\\ \dfrac{2H(P\cap Q)}{H(P)+H(Q)} & otherwise.\end{cases} \qquad (7)$$

Finally, the PMI (point-wise mutual information) is defined as:

$$WebPMI(P,Q)=\begin{cases}0 & if\ H(P\cap Q)\le c\\ \log_2\left(\dfrac{\frac{H(P\cap Q)}{N}}{\frac{H(P)}{N}\frac{H(Q)}{N}}\right) & otherwise\end{cases} \qquad (8)$$

H (P) denotes the page count for the query *P*. If the page counts for the query *P* and *Q*, that is *H(P ∩ Q),* is less than *c* ,we consider semantic relatedness between *P* and *Q* is zero, we set *c*=5 in our experiment.

Where *N* is the number of documents indexed by the Web search engine. In the experiments we set $N = 10^{10}$, according to the number of indexed pages reported by Google.

C. Integrating Page Counts and Web-based Kernel Function Measuring Semantic Relatedness

Semantic relatedness and semantic similarity between words are two different concepts; however, there are some connections between them. The more semantic similarity between words, the more semantic relatedness. So we can use semantic similarity to get a better result in measuring semantic relatedness between words. At the same time, we use page counts measuring semantic relatedness, then we can propose a new method to measuring semantic relatedness which integrating page counts and web- based kernel function.

First of all, we get *K(x,y)* that means semantic similarity between *x* and *y* with web-based kernel function. Second, *F(x,y)* is the semantic relatedness between *x* and *y*, and we obtain *F(x,y)* based on page counts. At last, we measure semantic relatedness between *x* and *y* like this:

$$R(x,y)=\alpha K(x,y)+(1-\alpha)F(x,y) \qquad (9)$$

α is a parameter, and 0<= α <=1.

IV. EXPERIMENTS AND RESULTS

We evaluated the proposed method by comparing our results with the Miller-Charles benchmark dataset. There are two steps: first of all, we compare the relatedness scores produced by the proposed method, and get the Correlation and Spearman rank correlation coefficient; then we change the parameter for a better result of Spearman rank correlation coefficient.

A. The Benchmark Dataset

We choose the Miller-Charles dataset; a dataset of 30 word-pairs [8] rated by a group of 38 human subjects, as the benchmark Dataset. Because of the omission of two word pairs in earlier versions of WordNet, so we use only 28 pairs for evaluations. The score is from 1(no relatedness) to 4(perfect synonymy). Although Miller-Charles dataset rise in 1991, it is highly correlated with some other benchmark

TABLE II. SPEARMAN RANK CORRELATION COEFFICIENT OF THESE METHODS BASED ON MILLER-CHARLES' DATASET

Method	**Spearman rank correlation coefficient**
WebJaccard	0.3924
WebDice	0.3924
WebOverlap	0.3990
WebPMI	0.3990
Sahami	0.6091
Proposed method(α =0.6)	0.6344

datasets. Therefore, Miller-Charles ratings can be considered as a good benchmark for computing semantic relatedness measures.

B. *Semantic Relatedness Results*

We integrate Page Counts and Web-based Kernel Function to measure semantic relatedness, and the results show in table 1. In table 1, WebPMI get the highest correlation score among the measures based on Page Counts. Web-based Kernel Function method, which is proposed by Sahami, performs better than WebPMI. When we use the method that integrating Page Counts and Web-based Kernel Function, the correlation is 0.724, compared to using Page Counts or Web-based Kernel Function alone, it obtains a much better result.

We get the spearman rank correlation coefficient of each method. Table 2 is the Spearman rank correlation coefficient of these methods based on Miller-Charles' dataset. The value is small when using page counts alone. Sahami proposed method get a reasonable result. In our method, we use the WebPMI method because it gets the highest correlation score, WebPMI replace $F(x,y)$ in formula 8. *Fig. 1* is the correlation with the change of α . When α =0.6, we almost get the highest correlation.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a web based method for measuring semantic relatedness between words. We integrate Page Counts and Web-based Kernel Function for Measuring Semantic Relatedness, and find that the result is close to measuring semantic relatedness by human being.

But there are lots of works to do in the future. Disambiguation is a problem in our method, for example, "apple" means a technology company, a kind of fruit and some other meanings, and we cannot get the exact meaning when measuring semantic relatedness. So we may use knowledge base to solve this problem in the next work.

ACKNOWLEDGMENT

This work was supported by the NSF of China (No.90920005, No.61003192), the Major Project of State Language Commission in the Twelfth Five-year Plan Period (No.ZDI125-1), the Project in the National Science &Technology Pillar Program in the Twelfth Five-year Plan Period (No.2012BAK24B01), the Program of Introducing Talents of Discipline to Universities (No.B07042), the NSF of Hubei Province (No.2011CDA034) and the self-determined research funds of CCNU from the colleges' basic research and operation of MOE (No.CCNU10A02009 and No.CCNU10C01005).

REFERENCES

[1] A. Budanitsky and G. Hirst. Evaluating wordnet based measures of lexical semantic relatedness. Computational Linguistics, 32(1):13–47, 2006.

[2] Michael Strube and Simon Paolo Ponzetto. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In AAAI'06, Boston, MA, 2006.

[3] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. Proceedings of IJCAI, 1606-1611, 2007.

[4] XU Yun, FAN Xiaozhong, ZHANG Feng. Semantic Relevancy Computing Based on HowNet[J]. Transcations of Beijing Institute of Technology, 2005,25(5):411-414(Ch).

[5] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In Proc. of 15th International World Wide Web Conference, 2006.

[6] Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka. Measuring Semantic Similarity between Words Using Web Search Engines, Proceedings of the 16th international conference on World Wide Web, May 08-12, 2007, Banff, Alberta, Canada.

[7] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513-523, 1988.

[8] G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. Language and Cognitive Processes, 6(1):1–28, 1991.

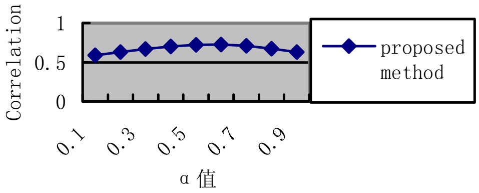

Figure 1. the correlation with the change of α