Exploiting Multiple Resources for Word-Phrase Semantic Similarity Evaluation

Xiaoqiang Jin, Chengjie Sun, Lei Lin, and Xiaolong Wang

School of Computer Science and Technology
Harbin Institute of Technology, China
{xqjin,chjsun,linl,wangxl}@insun.hit.edu.cn

Abstract. Previous researches on semantic similarity calculating have been mainly focused on documents, sentences or concepts. In this paper, we study the semantic similarity of words and compositional phrases. The task is to judge the semantic similarity of a word and a short sequence of words. Based on structured resource (WordNet), semi-structured resource (Wikipedia) and unstructured resource (Web), this paper extracts rich effective features to represent the word-phrase pair. The task can be treated as a binary classification problem and we employ Support Vector Machine to estimate whether the word and phrase is similar given a word-phrase pair. Experiments are conducted on SemEval 2013 Task5a. Our method achieves 82.9% in accuracy, and outperforms the best system (80.3%) that participates in the task. Experimental results demonstrate the effectiveness of our proposed approach.

Keywords: word-phrase Semantic Similarity, Support Vector Machine

1 Introduction

Semantic similarity calculation plays an important role in natural language processing, such as information retrieve, automatic question answering, word sense disambiguation and machine translation. Most previous studies about semantic similarity calculation mainly focus on documents, sentences or concepts, while ignore the fine-grained word-phrase semantic similarity which has been a research hot topic. In this paper, we study the semantic similarity of words and compositional phrases.

The word-phrase semantic similarity evaluation task is to judge the semantic similarity of a word and a short sequence of words. In each word-phrase pair, the word is a noun and the phrase is made up of an adjective and a noun pair. The task can be treated as a binary classification which judge the word-phrase pair is similar or not in semantic space. As the word-phrase pair lack of context, in this paper, we propose to extract rich features from multiple resources to represent word-phrase pair, including structured resource (WordNet), semi-structured resource (Wikipedia) and unstructured resource (Web). Totally, 27 features are obtained. Through feature selection, we finally get the best feature subset. Then we use a supervised learning algorithm—Support Vector Machine

(SVM) to judge whether the word-phrase pair is similar or not in semantic space. We conduct experiments on SemEval 2013 Task5a. Our method achieves 82.9% in accuract beyond the best system (80.3%) that participates in the task. Experimental results demonstrate the effectiveness of our approach.

2 Related Work

According to the technology used, semantic similarity mainly could be divided into three categories: purely statistical techniques , knowledge-based techniques and hybrid techniques[10]. The researching objects have been mainly focused on article, sentence and concept.

Vector Space Model (VSM) is a purely statistical technique, which use the distributional hypothesis [2–5] that words that occur in similar contexts tend to have similar meanings. The distributional hypothesis could be explained as that statistical pattern of human word usage can be used to figure out what people mean. When calculate word-word semantic similarity, VSM represents the word as a point of the high dimension of the text by counting the frequency of the word. Then the semantic similarity could be calculated by comparing the distance of the point. VSM has been widely used in the information retrieval area. VSM has been modified to calculate the semantic similarity between words,documents and patterns[1]. Many methods have improved VSM , Deerwester et al. (1990)[6] used truncated Singular Value Decomposition (SVD) on the term-document matrix to improve similarity measurements. Landauer and Dumais (1997)[7] applied truncated SVD to word semantic similarity, achieving human-level scores on multiple-choice synonym questions from the Test of English as a Foreign Language (TOEFL). Truncated SVD applied to document semantic similarity is called Latent Semantic Indexing (LSI), but it is called Latent Semantic Analysis (LSA) when applied to word semantic similarity.

Web-based methods also use purely statistical techniques. It bases on the huge Web corpus and usually uses search engine to get the statistical information. Based on the AltaVista search engine, Turney P (2001) [21] used pointwise mutual information(PMI) to recognize synonyms. Hsin-Hsi et al(2006) [22] proposed a web search with double checking model to explore the web as a live corpus. GANG LU et al(2010) [23] proposed a semantic similarity measurement method using the information page count and snippets. Danushka Bollegala et al(2008) [24] proposed an approach to compute semantic similarity using automatically extracted lexical-syntactic patterns from text snippets.

WordNet based methods are based on the knowledge. It encodes the important relations between the words such as synonymy, hypernymy and meronymy. In WordNet text semantic similarity is mapped to word sense traced as the concept. Methods [13–18] based on WordNet employ the structure of the WordNet, sometimes combing the information content.

Wiki-based methods are based on knowledge-Wikipedia. Like WordNet based methods, Wiki-based methods [9–12] either use the content in the page or the structure between the page. Explicit semantic analysis [9, 10] (ESA) treats each article in Wikipedia as a concept and map any text into the high dimension of concept, besides, ESA has also measured the link between the pages. Hybrid methods have used both the statistical techniques and knowledge-based techniques. Mihalcea et al (2006) [25] and Bar et al (2012) [26] used hybrid method to measure text semantic similarity.

Despite many methods have been used to calculate the semantic similarity, different methods focus on different lengths of texts and the effect of each semantic similarity is mainly dependent on the resource. There are three main resources frequently used: structured resource, semi structured resource and unstructured resource. VSM and ESA perform better in long text while WordNet and Web-based methods in word. Structured resource(e.g. WordNet) could provide more information about semantic similarity but a lower coverage, Besides it is difficult to update. Unstructured resource(e.g. Web) have a higher coverage but provide less information. Semi structured resource(e.g. Wikipedia) combine the advantages of structured resource and unstructured resource. But at the same time, it gets the drawbacks of them. Though the length of phrase is shorter than long text, we can still treat a phrase as a long text. Besides the semantic similarity of phrase can be calculated by word-word semantic similarity. To obtain the advantage of multiple resources and methods, we use multiple semantic similarities as the features of word-phrase pair and employ a supervised learning algorithm to calculate the semantic similarity of word-phrase.

3 Method

The word-phrase semantic similarity evaluation task is to judge whether the word and phrase in a given pair is similar in semantic space or not. We treat the task as a binary classification problem and employ SVM to solve it. The representation of word-phrase pair has a significant influence on the final performance. Therefore we focus on the feature crafting. In consideration of the lack of sufficient context information, we propose to exploit three kinds of resources, i.e. structured resource (Wordnet), semi-structured resource (Wikipedia) and unstructured resource (Web), to extract rich features for word-phrase pair representation. To combine the features from different resources, we employ a feature selection algorithm to select effective features. In the following, we first introduce three kind of features, and then give the details of the feature selection algorithm.

3.1 WordNet Based features

Adjective in English mostly act as a modifier, that is to say we can neglect the adjective in phrase without losing much information. Besides WordNet organizes same part of speech in a tree, it performs worse when used to measure the semantic similarity of words which have different part of speeches. Here to simplify the method to get the features, we use word-word semantic similarity as the feature instead of word-phrase semantic similarity. Many methods have been proposed to calculate word-word semantic similarity using WordNet. These methods are mainly based on structure in the WordNet or information content. Since a word in WordNet is represented by its synsets[1], we cannot calculate the word-word semantic similarity directly. Usually, we can use the following formula to get the word-word semantic similarity, by selecting for any given pair of words those two meanings that lead to the highest concept-concept semantic similarity.

$$Sim(w_1, w_2) = max_{c_1 \in SC_1 c_2 \in SC_2} Sim(c_1, c_2) \tag{1}$$

where w_1 and w_2 are the words which need to calculate the semantic similarity and w_1 and w_2 have the same part of speech, SC_1 and SC_2 are the synsets of w_1 and w_2 in the WordNet.

Here we give a short description of methods that are frequently used. Path semantic similarity method proposed by S&P[13] is defined as:

$$Sim_{S\&P}(c_1, c_2) = 2 * depthMax - len(c_1, c_2) \tag{2}$$

Where the $depthMax$ is the maximum depth of the taxonomy, $len(c_1, c_2)$ is the length of the shortest path between the concepts c_1 and c_2 using node counting.

Wup (Wu & Palmer 1994)[14] semantic similarity metric use the length of two given concepts in the WordNet taxonomy and the depth of the least common subsume(LCS).

$$Sim_{W\&P} = \frac{2 * depth(msc(c_1, c_2))}{len(c_1, c_2) + 2 * depth(msc(c_1, c_2))} \tag{3}$$

Lch(Leacock& Chodorow)[15] semantic similarity method is described as:

$$Sim_{L\&C} = -log(\frac{len(c_1, c_2)}{2 * depthMax}) \tag{4}$$

Res semantic similarity method introduced by Resnik (Resnik 1995)[16] use the information content(IC) of the LCS of the two concepts.

$$Sim_{res}(c_1, c_2) = IC(msc(c_1, c_2)) \tag{5}$$

Lin semantic similarity method[17], building on the Resnik's method ,adds a normalization factor consisting of the information content of the two input concepts:

$$Sim_{Lin}(c_1, C_2) = \frac{2 * IC(msc(c_1, c_2))}{IC(c_1) + IC(c_2)} \tag{6}$$

Jcn semantic similarity[18] is described as the following:

$$Sim_{jcn} = \frac{1}{IC(c_1) + IC(c_2) - 2 * IC(msc(c_1, c_2))} \tag{7}$$

We use six methods above to calculate the semantic similarity. The metrics can be easily achieved by using the WordNet-based implementation NLTK package[2].

[1] Sometimes synset is also called word concept.

[2] This package is implemented by Python, one can easily get the tool at the web site:http://www.nltk.org/

3.2 Wikipedia Based features

We get the Wikipedia based features mainly using the method Explicit Semantic Analysis (ESA) and Vector Space Model (VSM). ESA represents meaning of text in a high-dimensional space of concepts derived from Wikipedia, thus the meaning of any text can be represented in terms of Wikipedia-based concepts. The semantic similarity of the text can be valued by comparing the Wikipedia-based concepts vectors. Since ESA can process the text of arbitrary length, we can easily get the word-phrase semantic similarity. VSM builds a matrix of Term Frequency-Inverse Document Frequency (TF-IDF), in which rows correspond to terms and columns correspond to documents in the Wikipedia. Each word can be represented by a vector in the matrix of TF-IDF. We can get the word-word semantic similarity by comparing the vector in terms of the word. In order to get word-phrase semantic similarity, we use two methods. The first one is the same as the WordNet feature extraction, we use noun word-noun word semantic similarity to represent the word-phrase semantic similarity. The second one counts each word's weight contributing to the text semantic similarity, it use word-word semantic similarity to represent word-phrase semantic similarity. The following is the formula[25] that we use to get the semantic similarity:

$$\begin{aligned} Sim(T_1, T_2) = \frac{1}{2} * (&\frac{\sum\limits_{w \in \{T_1\}} maxsim(w, T_2) * idf(w)}{\sum\limits_{w \in \{T_1\}} idf(w)} \\ &+ \frac{\sum\limits_{w \in \{T_2\}} maxsim(w, T_1) * idf(w)}{\sum\limits_{w \in \{T_2\}} idf(w)}) \end{aligned} \tag{8}$$

Where $maxsim(w, T)$ is the highest semantic similarity between word w and $w^{'}$ which is a word in text T. The Formula (8) builds a bridge from word-word semantic similarity to text-text semantic similarity when calculating the semantic similarity. In this paper, we use the Wikipedia snapshot as November 1, 2012 to implement our ESA and VSM model. Following procedures have been used to process the Wikipedia XML dump:

1. Turn the Wikipedia dump format from XML to text using the tool Wikipedia_Extractor[3];
2. Stem the word[4] and change the capital case to lower case;
3. Filter out the concept[5] which has fewer than 200 words;
4. Filter out non-English characters and numbers;
5. Filter out the stop words;

[3] This is a tool to parse Wikipedia backup XML to separate text files, it could be downloaded at the website:http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

[4] This paper uses porter2 algorithm implemented by the stemming tool: https://pypi.python.org/pypi/stemming/1.0

[5] In Wikipedia an concept is same as a article.

After parsing the Wikipedia XML dump, we got 9.4 Gb of text in 1,542,428 articles and 3,323,004 distinct terms, which served for representing Wikipedia concepts as attribute vectors.

To compare the vector's similarity, we use four metrics: cosine, euclidean ,manhattan and jaccard. Giving two vector:

$$w = (w_1, w_2, ..., w_N)$$

$$v = (v_1, v_2, ..., v_N)$$

We can get the metrics using the following formula:

$$cos < w, v >= \frac{\sum_{i=1}^{N} w_i * v_i}{\sqrt{\sum_{i=1}^{N} w_i^2} * \sqrt{\sum_{i=1}^{N} v_i^2}} \tag{9}$$

$$eul < w, v >= \sqrt{\sum_{i=1}^{N} (w_i - v_i)^2} \tag{10}$$

$$man < w, v >= \sum_{i=1}^{N} |w_i - v_i| \tag{11}$$

$$jac < w, v >= \frac{\sum_{i=1}^{N} w_i * v_i}{\sum_{i=1}^{N} w_i^2 + \sum_{i=1}^{N} v_i^2 - \sum_{i=1}^{N} w_i * v_i} \tag{12}$$

There are three kinds of methods to get the semantic similarity vector. And we use four metrics to evaluate the distance between vectors. Totally, 12 features are obtained based on Wikipedia.

3.3 Web Based features

Web Based features use the word co-occurrence[19]. The core idea is that "a word is characterized by the company it keeps" [20]. Since the Web page is too huge to get the date local for us, we mostly use the search engine to get the statistical information about the term in the Web. So many metrics could be used to measure the degree of the word co-occurrence. In this paper, we get the word statistical information through AltaVista and use Pointwise Mutual Information (PMI), Jaccard coefficient and Dice coefficient to measure the word co-occurrence. If we define the $p(query)$ as the following[21]:

$$p(query) \approx \frac{hits(query)}{N} \tag{13}$$

Giving two query q_1, q_2 , the definitions of the metrics could be represented as follow formula:

$$WebDice(q_1, q_2) = log_2(\frac{2 * hits(q_1 \wedge q_2)}{hits(q_1) + hits(q_2)}) \tag{14}$$

$$WebPMI(q_1, q_2) = log_2(\frac{hits(q_1 \wedge q_2)/N}{(hits(q_1)/N) * (hits(q_2)/N)} \tag{15}$$

$$WebJaccard(q_1, q_2) = log_2(\frac{2 * hits(q_1 \wedge q_2)}{hits(q_1) * hits(q_2) - hits(q_1 \wedge q_2)}) \tag{16}$$

Where hits(q) is the number of the web page returning by the search engine containing the query q; N is the number of the total web pages, usually $N = 10^{13}$[21].

From the four different types of queries suggested by Turney (2001)[21], we are using the NEAR query (co-occurrence within a ten-word window), which is a balance between accuracy (results obtained on synonymy tests) and efficiency (number of queries to be run against a search engine)[25]. Specifically, the co-occurrence query could be defined as the follow to collect counts from the AltaVista search engine.

$$hits(q_1 \wedge q_2) = hits(q_1 NEAR\ q_2) \tag{17}$$

This paper use three ways to get the web based features:

1. Use noun word-noun word semantic similarity to represent word-phrase semantic similarity.
2. Use word-word semantic similarity to get the semantic similarity of word-phrase, which can be achieved by formula(8).
3. Treat the phrase as 'word', use the Web-based method to get the semantic similarity directly.

Here three metrics are used to evaluate the word co-occurrence, finally we achieve 9 features based on Web.

3.4 Feature Combination

Based on three resources: Wikipedia, WordNet and Web corpus, We total get 27 features. According to the method used to get the semantic similarity of word-phrase, we classify the features into three categories. To make the feature notations more readable, we use a capital letter as a prefix before method to represent the category and the letters after '-' to denote the metric we use.

1. Use word-to-word semantic similarity to represent word-to-phrase semantic similarity, here we use the prefix—'W' before methods to represent this category:

   (a) WVSM-COS,WVSM-EUL,WVSM-JAC,WVSM-MAN[6]
   (b) WIR-JAC,WIR-DICE,WIR-PMI
2. Use noun word-to noun word semantic similarity to represent word-to-phrase semantic similarity,here we use the prefix—'N' before methods to represent this category:
   (a) NLIN,NWUP,NRES,NPATH,NLCH,NJCN
   (b) NESA-COS,NESA-EUL,NESA-JAC,NESA-MAN[7]
   (c) NIR-JAC,NIR-DICE,NIR-PMI
3. Directly calculate word-phrase semantic similarity, here we use prefix—'D' before methods to represent this category:
   (a) DESA-COS,DESA-EUL,EDSA-JAC,DESA-MAN
   (b) DIR-JAC,DIR-DICE,DIR-PMI

Here we treat each semantic similarity as feature of word-phrase pair, thus each instance can be represented as a vector of 27 features.

We use two ways to combine feature. One way is to combine all the features to calculate the semantic similarity and the other is through feature selection. We use the symbols FAll and FSubset to represent the ways of feature combination separately.

Here, we use a heuristic search procedure—forward search [27] to find a good feature subset. We employ SVM[8] as the evaluation function of feature selection. The forward search can be represented as follow:

1. Initialize $\mathcal{F} = \emptyset$.
2. Repeat
   (a) For $i = 1, 2, ..., n$ if $i \notin \mathcal{F}$,let $\mathcal{F}_i = \mathcal{F} \cup \{i\}$, and use 5-flod cross validation to evaluate features $\mathcal{F}_i$.(Train the learning algorithm(SVM) using only the features in $\mathcal{F}_i$,and estimate its accuracy.)
   (b) Set $\mathcal{F}$ to be the best features subset found on step (a)
3. Select and output the best feature subset that was evaluated during the entire search procedure.

The outer loop of the algorithm can be terminated either when $\mathcal{F} = \{1, ...n\}$ is the set of all features, or when $|\mathcal{F}|$ exceeds some pre-set threshold(corresponding to the maximum number of features that you want the algorithm using).

Through feature selection, we get the best feature subset. The best feature subset—FSubset is as follow:

FSubset = {NESA-JAC,WVSM-EUC,WVSM-MAN,DESA-EUC,DESA-JAC, NIR-PMI,DIR-DICE, WIR-JAC,WIR-PMI,NLCH,NRES }

[6] When use word-to-word semantic similarity to calculate word-phrase semantic similarity, ESA and VSM are the same.
[7] When use noun word-to-noun word semantic similarity to calculate word-phrase semantic similarity, ESA and VSM are the same.
[8] This paper use the open source tool libsvm:http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

4 Experiments

4.1 Data Set

The data we use are from SemEval2013-Task5a[9]. The train data set consists of 5861 negative instances and 5861 positive instances. In the test set, 3906 instances are provided. The format of instance (e.g. time particular vs moment) in train and test data are the same. The first part of the instance is a noun and the second is a phrase which consists of an adjective and a noun.

4.2 Results and Discussion

To see the performance of each previous semantic similarity on word-phrase, we use each semantic similarity of word-phrase as the feature vector of word-phrase. Then we use the SVM to get the final results. This paper use three main metrics in SemEval2013-Task5a to evaluate results. According to the resources, we organize the results into three tables 1,2,3.

Table 1. The results of Wikipedia-based methods

Feature Symbol	Accuracy	Precision	Recall
NESA-COS	0.636	0.803	0.361
NESA-EUC	0.514	0.593	0.090
NESA-MAN	0.500	1.000	0.001
NESA-JAC	0.556	0.853	0.136
WVSM-COS	0.661	0.826	0.407
WVSM-EUC	0.518	0.573	0.141
WVSM-MAN	0.507	0.935	0.015
WVSM-JAC	0.582	0.868	0.195
DESA-COS	0.663	0.813	0.424
DESA-EUC	0.522	0.574	0.172
DESA-MAN	0.500	1.000	0.001
DESA-JAC	0.586	0.849	0.209

The table 1 shows the results of Wikipedia-based methods, as we can see, DESA* methods perform better than others. It can be explained as that ESA treats the phrase as a whole while others whether measure the partial information or measure the semantic similarity indirectly. Among all the metrics used to compare vector's similarity, cosine and jaccard have a better result in all the methods.

The table 2 reports the results of Web-based method. Based on the results, We find that most WIR* methods perform better than NIR* methods, We explain this as that though adjacent pay less contributions to semantic similarity than noun,it still has some contributions. DIR* methods preform worse than

[9] http://www.cs.york.ac.uk/semeval-2013/task5/index.php?id=full

Table 2. The results of Web-based methods

Feature Symbol	Accuracy	Precision	Recall
DIR-JAC	0.644	0.915	0.318
DIR-DICE	0.645	0.915	0.320
DIR-PMI	0.641	0.913	0.312
WIR-JAC	0.754	0.779	0.709
WIR-DICE	0.754	0.778	0.712
WIR-PMI	0.676	0.704	0.607
NIR-JAC	0.735	0.725	0.758
NIR-DICE	0.735	0.727	0.754
NIR-PMI	0.731	0.725	0.746

WIR* methods and NIR* methods, we conjecture that there is a main reason for the phenomena : this is due to the rare occurrence of phrase in the Web.

Table 3. The results of WordNet-based methods

Feature Symbol	Accuracy	Precision	Recall
NWUP	0.730	0.868	0.544
NJCN	0.646	0.922	0.320
NLCH	0.738	0.785	0.655
NLIN	0.673	0.841	0.428
NPATH	0.718	0.938	0.467
NRES	0.719	0.824	0.557

When we compare the data between table1, table2 and table3, we note WordNet-based methods perform better than other resources. It is because WordNet has much more information about semantic similarity than others.

Table 4. The result of top three in the SemEval 2013 task5a & feature combinations

Feature Symbol	Accuracy	Precision	Recall
Top three in the SemEval2013-task5a			
1st	0.803	0.837	0.752
2nd	0.794	0.867	0.685
3rd	0.794	0.856	0.707
Feature Combination			
FAll	0.814	0.851	0.760
FSubset	0.829	0.872	0.771

Table 4 shows the results of feature combination, besides the results of top three in SemEval 2013 task5a are also given. It is obvious that previous semantic similarity method is effective when we employ them to calculate the semantic similarity of word-phrase. We find that the result of FAll is significantly improved and FSubset preforms even better. Different features of semantic similarity based on different resources may figure out the different aspects of semantic similarity of word-phrase. That is key point to explain why FAll obtains a better result. Because of the information redundancy between features, that is why FSubset perform better than FAll. We also find that both the FAll and FSubset perform better than the top one in the Evaluating Phrasal Semantics task in three main metrics. That demonstrates the effectiveness of our approach.

5 Conclusion

In this paper we focus on the task to judge whether the word-phrase is similar or not in semantic space. Based on multiple resources, this paper extracts rich effective features to represent the word-phrase pair. After a feature combination, we employ SVM to estimate whether the word and phrase is similar or not in semantic space. Experimental results conducted on SemEval2013-Task5a's data set demonstrate the effectiveness of our approach.

Although the approach performs better than the best result in the Evaluating Phrasal Semantics task at SemEval-2013 in three main metrics, there is still room for improvement. There may be more effective features to calculate the semantic similarity. And other classifier excluding SVM, may have a better performance.

Acknowledgment

This work is supported by the Key Basic Research Foundation of Shenzhen (JC201005260118A) and National Natural Science Foundation of China (61100094 & 61300114).The authors are grateful to the anonymous re-viewers for their constructive comments. Special thanks to Yaming Sun for insightful suggestions.

References

1. Turney P D, Pantel P. From frequency to meaning: Vector space models of semantics[J]. Journal of artificial intelligence research, 2010, 37(1): 141-188.
2. Wittgenstein, L. (1953). Philosophical Investigations. Blackwell. Translated by G.E.M.Anscombe.
3. Harris, Z. (1954). Distributional structure. Word, 10 (23), 146-162
4. Weaver, W. (1955). Translation. In Locke, W., & Booth, D. (Eds.), Machine Translation of Languages: Fourteen Essays. MIT Press, Cambridge, MA.
5. Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. In Studies in Linguistic Analysis, pp. 1-32. Blackwell, Oxford.
6. Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science (JASIS), 41 (6), 391-407.
7. Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104 (2), 211-240.

8. Han E H S, Karypis G. Centroid-based document classification: Analysis and experimental results[M]. Springer Berlin Heidelberg, 2000.
9. Gabrilovich E, Markovitch S. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis[C]//IJCAI. 2007, 7: 1606-1611.
10. Gabrilovich E, Markovitch S. Wikipedia-based semantic interpretation for natural language processing[J]. Journal of Artificial Intelligence Research, 2009, 34(2): 443.
11. Strube M, Ponzetto S P. WikiRelate! Computing semantic relatedness using Wikipedia[C]//AAAI. 2006, 6: 1419-1424.
12. Witten I, Milne D. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links[C]//Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA. 2008: 25-30.
13. Resnik P. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language[J]. arXiv preprint arXiv:1105.5444, 2011.
14. Wu Z, Palmer M. Verbs semantics and lexical selection[C]//Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 1994: 133-138.
15. Leacock C, Chodorow M. Combining local context and WordNet similarity for word sense identification[J]. WordNet: An electronic lexical database, 1998, 49(2): 265-283.
16. Resnik P. Using information content to evaluate semantic similarity in a taxonomy[J]. arXiv preprint cmp-lg/9511007, 1995.
17. Lin D. An information-theoretic definition of similarity[C]//ICML. 1998, 98: 296-304.
18. Jiang J J, Conrath D W. Semantic similarity based on corpus statistics and lexical taxonomy[J]. arXiv preprint cmp-lg/9709008, 1997.
19. Manning, C.D., Schtze, H.: Foundations of Statistical Natural Language Processing. Cambridge, Massachusetts: MIT Press (1999).
20. Firth, J.R.: A Synopsis of Linguistic Theory 1930-1955. In Studies in Linguistic Analysis, pp. 1-32. Oxford: Philological Society (1957). Reprinted in F.R. Palmer (ed.), Selected Papers of J.R. Firth 1952-1959, London: Longman (1968).
21. Turney P. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL[J]. 2001.
22. Chen H H, Lin M S, Wei Y C. Novel association measures using web search with double checking[C]//Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2006: 1009-1016.
23. Lu G, Huang P, He L, et al. A new semantic similarity measuring method based on web search engines[J]. WSEAS Transactions on Computers, 2010, 9(1): 1-10.
24. Bollegala D, Matsuo Y, Ishizuka M. Measuring semantic similarity between words using web search engines[J]. www, 2007, 7: 757-766.
25. Mihalcea R, Corley C, Strapparava C. Corpus-based and knowledge-based measures of text semantic similarity[C]//AAAI. 2006, 6: 775-780.
26. Bar, D., Biemann, C., Gurevych, I., & Zesch, T. (2012, June). Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (pp.435-440). Association for Computational Linguistics.
27. Andrew Ng, Regularization and model selection, CS 229 Machine Learning Course Materials pp.4-5.