COMPUTING SEMANTIC SIMILARITY MEASURE BETWEEN WORDS USING WEB SEARCH ENGINE

Pushpa C N, Girish S, Nitin S K, Thriveni J, Venugopal K R and L M Patnaik

[1]Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering, Bangalore
[2] Indian Institute of Science, Bangalore
pushpacn@gmail.com

ABSTRACT

Semantic Similarity measures between words plays an important role in information retrieval, natural language processing and in various tasks on the web. In this paper, we have proposed a Modified Pattern Extraction Algorithm to compute the supervised semantic similarity measure between the words by combining both page count method and web snippets method. Four association measures are used to find semantic similarity between words in page count method using web search engines. We use a Sequential Minimal Optimization (SMO) support vector machines (SVM) to find the optimal combination of page counts-based similarity scores and top-ranking patterns from the web snippets method. The SVM is trained to classify synonymous word-pairs and non-synonymous word-pairs. The proposed Modified Pattern Extraction Algorithm outperforms by 89.8 percent of correlation value.

KEYWORDS

Information Retrieval, Semantic Similarity, Support Vector Machine, Web Mining, Web Search Engine, Web Snippets.

1. INTRODUCTION

Search engines have become the most helpful tool for obtaining useful information from the Internet. The search results returned by the most popular search engines are not satisfactory. The study of semantic similarity between words has been an integral part of information retrieval, natural language processing and in various tasks on the web such as relation extraction, community mining, document clustering, automatic meta data extraction and Web mining applications such as, community extraction, relation detection, and entity disambiguation in information retrieval. Semantic similarity is a concept whereby a set of terms within term lists are assigned a metric based on the likeness of their meaning.

Due to the vastness of the web, it is impossible to analyse each document separately, hence Web search engines provide the perfect interface for this vast information. Page count of a query term will give an estimate of the number of documents or web pages that contain the given query term. A web snippet is one which appears below the searched documents and is a brief window of text that is searched around the query term in the document. Processing snippets is possible for

David C. Wyld (Eds) : ICCSEA, SPPR, CSIA, WimoA - 2013
pp. 135–142, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3514

measuring semantic similarity but it has the drawback of downloading a large number of web pages which consumes time and most of the search engine algorithms use a page rank algorithm, hence only the top ranked pages will have properly processed snippets. Hence there is no guarantee that all the information we need is present in the top ranked snippets.

Page count between two objects is accepted globally as the relatedness measure between them. For example, the page count of the query *"apple"* AND "*computer*" in Google is 977,000,000 whereas the same for "*banana*" AND "*computer*" is only 60,200,000 [as on 20 December 2012]. The 16 times more than numerous page counts for "*apple*" AND "*computer*" indicate that "*apple*" is more semantically similar to "*computer*" than is "*banana*". Page counts and snippets are two useful information sources provided by most Web search engines.

Motivation: The search results returned by the most popular search engines are not satisfactory due to the vastly numerous documents and the high growth rate of the Web, it is time consuming to analyse each document separately. Information retrieval such as search engines, the semantic similarity is the main problem to retrieve all the documents that are semantically related to the queried term by the user. Page counts of a query and web snippets are two useful information sources provided by most Web search engines. In dictionary the semantic similarity between words is solved, but when it comes to web, accurately measuring the semantic similarity between words is a very challenging task.

Contribution: We propose a Modified Pattern Extraction Algorithm to find the supervised semantic similarity measure between words by combining both page count method and web snippets method. Four association measures including variants of Web Dice, Web Overlap Ratio, Web Jaccard, and WebPMI are used to find semantic similarity between words in page count method using web search engines. The proposed approach aims to improves the correlation value compared to the existing methods.\\

Organization: The remainder of the paper is organized as follows: Section 2 reviews the related work of the semantic similarity measures between words, Section 3 gives the problem definition, Section 4 explains the architecture of the system, Section 5 gives Page-count-based Co-occurrence Measures, and Section 6 gives the Modified Pattern Extraction Algorithm [MPEA]. The implementation and the results of the system are described in Section 7 and Conclusions are presented in Section 8.

2. RELATED WORK

Mehran Sahami et al., [1] proposes a novel method for measuring the similarity between short text snippets by leveraging web search results to provide greater context for the short texts. The proposed method captures more of the semantic context of the snippets rather than simply measuring their term-wise similarity. Hsin-Hsi Chen et al., [2] proposed novel model is a Web Search with Double Checking (WSDC) web search with double checking model to explore the web as a live corpus and is used to analyze snippets.

Rudi L. Cilibrasi et al., [3] presents the words and phrases acquire meaning from the way they are used in society, from their relative semantics to other words and phrases. It is a new theory of similarity between words and phrases based on information distance and Kolmogorov complexity. The method is applicable to all search engines and databases. Dekang Lin et al., [4] proposed that, bootstrapping semantics from text is one of the greatest challenges in natural language learning. They defined a word similarity measure based on the distributional pattern of words. Jian Pei et al.,[5] proposed a projection-based, sequential pattern-growth approach for efficient mining of sequential patterns. Jiang et al., [6] combines a lexical taxonomy structure with corpus statistical information so that the semantic distance between nodes in the semantic space constructed by the taxonomy can be better quantified with the computational evidence derived from a distributional analysis of corpus data.

Philip Resnik et al., [7] - [8] presents measure of semantic similarity in an is-a taxonomy, based on the notion of information content. Bollegala et al., [9] proposed a method which exploits the page counts and text snippets returned by a Web search engine. Ming Li et al., [10] proposed a metric based on the non-computable notion of Kolmogorov computable distance and called it the similarity metric. General mathematical theory of similarity that uses no background knowledge or features specific to an application area. This literature survey proved the fact that semantic similarity measures play an important role in various fields. Thus there is need for more efficient system to find semantic similarity between words.

3. PROBLEM DEFINTION

Given two words A and B, we model the problem of measuring the semantic similarity between A and B, as a one of constructing a function semanticsim (A, B) that returns a value in the range of 0 and 1. If A and B are highly similar (e.g. synonyms), we expect semantic similarity value to be closer to 1, otherwise semantic similarity value to be closer to 0. We define numerous features that express the similarity between A and B using page counts and snippets retrieved from a web search engine for the two words. Using this feature representation of words, we train a two-class Support Vector Machine (SVM) to classify synonymous and non-synonymous word pairs. Our objective is to find the semantic similarity between two words and improves the correlation value.

4. SYSTEM ARCHITECTURE

The outline of the proposed method for finding the semantic similarity using web search engine results is as shown in Fig. 1.

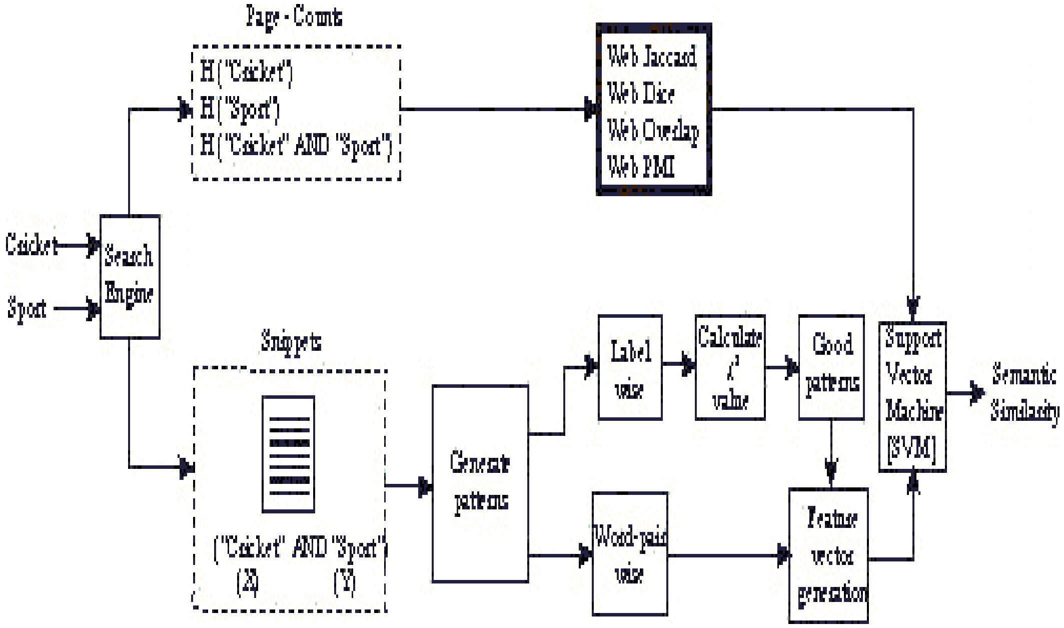

Fig. 1. System Architecture

When a query *q* is submitted to a search engine, web-snippets, which are brief summaries of the search results, are returned to the user. First, we need to query the word-pair in a search engine for example say we query $cricket$ and $sport$ in Google search engine. We get the page counts of the word-pair along with the page counts for individual words i.e. H(*cricket*), H(*sport*), H(*cricket AND sport*). These page counts are used to find the co-occurrence measures such as Web-Jaccard, Web-Overlap, Web-Dice a nd Web-PMI and store these values for future references. We collect the snippets from the web search engine results. Snippets are collected only for the query X and Y. Similarly, we collect both snippets and page counts for 200 word pairs. Now we need to extract patterns from the collected snippets using our Modified pattern extraction algorithm and to find the frequency of occurrence of these patterns.

The chi-square statistical method is used to find out the good patterns from the top 200 patterns of interest using the pattern frequencies. After that we integrate these top 200 patterns with the co-occurrence measures computed. If the pattern exists in the set of good patterns then we select the good pattern with the frequency of occurrence in the patterns of the word-pair else we set the frequency as 0. Hence we get a feature vector with 204 values i.e,. the top 200 patterns and four co-occurrence measures values. We use a Sequential Minimal Optimization (or SMO) support vector machines (SVM) to find the optimal combination of page counts-based similarity scores and top-ranking patterns. The SVM is trained to classify synonymous word-pairs and non-synonymous word-pairs. We select synonymous word-pairs and Non-synonymous word-pairs and convert the output of SVM into a posterior probability. We define the semantic similarity between two words as the posterior probability if they belong to the synonymous-words (positive) class.

5. PAGE-COUNT-BASED CO-OCCURRENCE MEASURES

We compute four popular co-occurrence measures; Jaccard, Overlap (Simpson), Dice, and Point wise Mutual Information (PMI), to compute semantic similarity using page counts.

Web Jaccard coefficient between words (or multi-word phrases) A and B, is defined as:

$$\mathrm{WebJaccard}(A,B) = \begin{cases} 0 & \text{if } H(A \cap B) \le c, \\ \dfrac{H(A \cap B)}{H(A) + H(B) - H(A \cap B)} & \textit{otherwise} \end{cases} \qquad (1)$$

Web Overlap is a natural modification to the Overlap (Simpson) coefficient, is defined as:

$$\mathrm{Web\ Overlap}(A,B) = \begin{cases} 0 & \text{if } H(A \cap B) \le c, \\ \dfrac{H(A \cap B)}{\min(H(A), H(B))} & \text{otherwise} \end{cases} \qquad (2)$$

WebDice is defined as:

$$\text{Web Dice}(A, B) = \begin{cases} 0 & \text{if } H(A \cap B) \le c, \\ \dfrac{2\,H(A \cap B)}{H(A) + H(B)} & \text{otherwise} \end{cases} \qquad (3)$$

Web PMI is defined as:

$$\text{Web PMI}(A, B) = \begin{cases} 0 & \text{if } H(A \cap B) \le c, \\ \log_{2}\left(\dfrac{\frac{H(A \cap B)}{N}}{\frac{H(A)}{N}\,\frac{H(B)}{N}}\right) & \textit{otherwise} \end{cases} \qquad (4)$$

6. ALGORITHM

The proposed Modified Pattern Extraction Algorithm is used to measure the semantic similarity between words is as shown in the Table 1.

```
Step 1: Read each snippet S, remove all the non-ASCII
          Character and store it in database.
Step 2:  for each snippet S do
                    if word is same as A then      Replace A by X
                    if word is same as B then      Replace B by Y
         end for
Step 3:  for each snippet S do
               if  X € S then goto Step 4
               if  Y € S then goto Step 9
         end for
Step 4:  if  Y or  Number of  words > Max. length L then
                stop the sequence seq.
         end if
Step 5:  for each seq do
              Perform stemming operation.
         end for
Step 6:  Form the sub-sequences of the sequence such that each
          sub-sequence contains [X . . . Y . .].
Step 7:  foreach subseq do
              if subseq is same as existing pattern and  unique then
                    list_ pat = list_ pat + subseq
                    freq _pat = freq _pat + 1
              end if
         end for
Step 8:   if   length exceeds L then
```

```
                Discard the pattern until you find an X or Y.
            end if
Step 9:  if  you encounter Y then
                goto Step 4.
         end if
```

Given two words A and B, we query a web search engine using the wildcard query A * * * * * B and download snippets. The * operator matches one word or none in a web page. Therefore, our wildcard query retrieves snippets in which A and B appears within a window of seven words. Because a search engine snippet contains 20 words on an average and we assume that the seven word window is sufficient to cover most relations between two words in snippets. The algorithm which is described in the Table 1 shows that how to extract the patterns and the frequency of the patterns.

The Modified pattern extraction algorithm as described above yields numerous unique patterns. Of those patterns only 80% of the patterns occur less than 10 times. It is impossible to train a classifier with such numerous parse patterns. We must measure the confidence of each pattern as an indicator of synonymy that is, most of the patterns have frequency less than 10 so it is very difficult to find the patterns which are significant so, we have to compute their confidences as to arrive at the significant patterns. We compute chi-square value to find the confidence of each pattern.

The chi-square value is calculated by using the formula given below:

$$\mathcal{X}^2 = \frac{(P + N)\,(p_v\,(N - n_v) - n_v(P - p_v))^2}{PN\,(p_v + n_v)(P + N - p_v - n_v)} \qquad (5)$$

Where,

P and N are the Total frequency of synonymous word pair patterns and non-synonymous word pair patterns, p_v and n_v are frequencies of the pattern v retrieved from snippets of synonymous and non-synonymous word pairs respectively.

7. Implementation and Results

We have implemented this in Java programming language and used an Eclipse is an extensible open source IDE (integrated development environment) [11]. We query for A AND B and collect 500 snippets for each word pair and for each pair of words (A, B) create a feature vector as explained above. Convert the feature vectors of all the word-pairs into a .CSV (Comma Separated Values) file. The generated .CSV file is fed to the SVM classifier which is inbuilt in Weka software. This classifies the values and gives a similarity score for the word-pair in between 0 and 1. In order to test our system, we selected the standard Miller-Charles dataset, which is having 28 word-pairs. The proposed algorithm outperforms by 89.8 percent of correlation value, as illustrated in Table 2.

Table 2. Semantic Similarity on Miller-Charles Dataset

Word Pair	Miller-Charles	Web Jaccard	Web Dice	Web Overlap	Web PMI	Bollegala Method	MPEA
automobile-car	1.00	0.65	0.66	0.83	0.43	0.92	0.98
journey-voyage	0.98	0.41	0.42	0.16	0.47	1	0.93
gem-jewel	0.98	0.29	0.3	0.07	0.69	0.82	0.98
boy-lad	0.96	0.18	0.19	0.59	0.63	0.96	0.95
coast-shore	0.94	0.78	0.79	0.51	0.56	0.97	0.98
asylum-madhouse	0.92	0.01	0.01	0.08	0.81	0.79	0.86
magician-wizard	0.89	0.29	0.03	0.37	0.86	1	0.94
midday-noon	0.87	0.1	0.1	0.12	0.59	0.99	0.71
furnace-stove	0.79	0.39	0.41	0.1	1	0.88	0.94
food-fruit	0.78	0.75	0.76	1	0.45	0.94	0.97
bird-cock	0.77	0.14	0.15	0.14	0.43	0.87	0.91
bird-crane	0.75	0.23	0.24	0.21	0.52	0.85	0.65
implement-tool	0.75	1	1	0.51	0.3	0.5	0.74
brother-monk	0.71	0.25	0.27	0.33	0.62	0.27	0.54
crane-implement	0.42	0.06	0.06	0.1	0.19	0.06	0.18
brother-lad	0.41	0.18	0.19	0.36	0.64	0.13	0.68
car-journey	0.28	0.44	0.45	0.46	0.2	0.17	0.26
monk-oracle	0.27	0	0	0	0	0.8	0.7
food-rooster	0.21	0	0	0.41	0.21	0.02	0.36
coast-hill	0.21	0.96	0.97	0.26	0.35	0.36	0.18
forest-graveyard	0.2	0.06	0.06	0.23	0.49	0.44	0.77
monk-slave	0.12	0.17	0.18	0.05	0.61	0.24	0.08
coast-forest	0.09	0.86	0.87	0.29	0.42	0.15	0.07
lad-wizard	0.09	0.06	0.07	0.05	0.43	0.23	0.03
cord-smile	0.01	0.09	0.1	0.02	0.21	0.01	0.03
glass-magician	0.01	0.11	0.11	0.4	0.6	0.05	0.04

rooster-voyage	0	0	0	0	0.23	0.05	0.06
noon-string	0	0.12	0.12	0.04	0.1	0	0.03
Correlation	**1.0**	**0.26**	**0.27**	**0.38**	**0.55**	**0.87**	**0.898**

7. Conclusions

Semantic Similarity measures between words plays an important role information retrieval, natural language processing and in various tasks on the web. We have proposed a Modified pattern extraction algorithm to extract numerous semantic relations that exist between two words and the four word co-occurrence measures were computed using page counts. We integrate the patterns and co-occurrence measures to generate a feature vector. These feature vectors are fed to a 2-Class SVM to classify the data into synonymous and non-synonymous classes. We compute the posterior probability for each word-pair which is the similarity score for that word-pair. The proposed algorithm outperforms by 89.8 percent of correlation value. Further, we are trying to calculate and compare the precision, recall and F-measure values of our methods with previous methods.

References

[1] Sahami, M., Heilman, T, (2006) “A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets”, In the Proceedings of 15th International Conference on World Wide Web (WWW 2006) Montreal, ACM New York, NY, USA, pp. 377–386.

[2] Chen H, Lin M & Wei Y, (2006) “ Novel Association Measures using Web Search with Double Checking”, International Committee on Computational Linguistics and the Association for Computational Linguistics, pp. 1009-1016 **.**

[3] Cilibrasi R & Vitanyi P, (2007) “ The google similarity distance”, IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 3, pp. 370-383.

[4] Lin D, (1998) “Automatic Retrieival and Clustering of Similar Words”, International Committee on Computational Linguistics and the Association for Computational Linguistics, pp. 768-774.

[5] Pei J, Han J, Mortazavi-Asi B, Wang J, Pinto H, Chen Q, Dayal U & Hsu M, (2004) “Mining Sequential Patterns by Pattern growth: the Prefix span Approach”, IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 11, pp. 1424-1440.

[6] Jay J Jiang & David W Conrath, (1997) “Semantic Similarity based on Corpus Statistics and Lexical Taxonomy”, International Conference Research on Computational Linguistics.

[7] Resnik P, (1995) “Using Information Content to Evaluate Semantic Similarity in a Taxonomy”, 14th International Joint Conference on Artificial Intelligence, Vol. 1, pp. 24-26.

[8] Resnik P, (1999) “Semantic Similarity in a Taxonomy: An Information based Measure and its Application to problems of Ambiguity in Natural Language”, Journal of Artificial Intelligence Research , Vol. 11, pp. 95-130.

[9] Danushka Bollegala, Yutaka Matsuo & Mitsuru Ishizuka, (2011) “A Web Search Engine-based Approach to Measure Semantic Similarity between Words”, IEEE Transactions on Knowledge and Data Engineering , Vol. 23, No.7, pp. 977-990.

[10] Ming Li, Xin Chen, Xin Li, Bin Ma, Paul M & B Vitnyi, (2004) “The Similarity Metric”, IEEE Transactions on Information Theory, Vol. 50, No. 12, pp. 3250-3264.

[11] http://onjava.com/onjava/2002/12/11/eclipse.html