Efficient Search Engine Approach for Measuring Similarity between words

Using Page Count and Snippets

P.Murugesan
PG student computer science and engineering
Indian Institute of Information Technology, Srirangam
Tiruchirappalli
gatemurugesan@gmail.com

K.Malathi
Faculty, Deportment of Information Technology
Indian Institute of Information Technology, Srirangam
Tiruchirappalli
mailtokmalathi@gmail.com

Abstract— **Web mining involve activities such as document clustering, community mining etc., to be performed on web. Such tasks need measuring semantic similarity between word. This helps in performing web mining activities easily in many applications. The accurate measures of semantic similarity between any two words is the difficult task. A new approach to measure similarity between words is based on text snippets and page count. These two measures are taken from the results of a search engine like Google. The lexical patterns are extracted from text snippets and word co-occurrence measures are defined using page count. The results of these two are combined. Moreover, the pattern clustering and pattern extraction algorithm are used to find various relationships between any two given words. Support Vector Machines is used to optimize the result. The empirical results reveal that the techniques are finding the best results that can be compared with human ratings and accuracy in web mining activity. Semantic similarity refers to the concept by which a set of document or words within the document are assigned a weight based on their meaning. The accurate measurement of such similarity plays an important role in Natural language Processing**.

***Index Terms*—Community mining, pattern clustering, text snippets, page count.** *(key words)*

I. INTRODUCTION

While searching a word in search engine it gives appropriate meaning. Normally most of the similar words and products have common names the search engine find difficult to identify what the user exactly needs. In this case by using clustering algorithm one can find the appropriate content for the searching word. Semantic similarity is a central concept that finds. Accurate measurement of semantic similarity between words is essential for various tasks such as, document clustering, information retrieval and synonym extraction. It should be able to understand the semantics or meaning of the words. But a computer being a syntactic machine, semantic associated with the words or terms is to be represented as syntax. In some instance the word "Apple" is some related to computer science as there is a company by name "Apple" which has been instrumental in bringing many computer hardware and software technologies. However, this word is ignored in some of the lexical dictionary as they consider as a fruit. As new words are created and many meanings are associated with the words, the lexical dictionary have proved to be inadequate to handle things when the words having different meanings and relationship with other words which are not yet updated in lexical dictionary.

II. RELATED WORK

Given taxonomy of word, a straightforward method is to calculate similarity between two words to find the length of the shortest path connecting the two word in the taxonomy[7]. If a word is polysemous, then multiple paths might exist between the two words. In such case, only the shortest path between any two senses of the words is considered for calculating similarity. A problem that is frequently acknowledge with this approach is that it relies on the notion that the links in the represent a uniform distance. Resnik proposed a similarity measure using information content[8]. The similarity between two concepts C1 and C2 in the taxonomy the maximum of the information content of the concept C that subsume both C1 and C2. The similarity between two words is defined as the maximum of the similarity between any concepts that the word belong to. He used WordNet as the taxonomy; information content is calculated using the Brown corpus[11]. Semantic similarity measures have been used the various applications in natural language processing such as word sense disambiguation, language modelling, synonym extraction, and automatic thesauri extraction[10]. Semantic similarity measure are important in many web related task. In query expansion, a user query is modified using synonymous words to improve the relevancy of the Search. One method to find appropriate word to include in a query is to compare the previous user queries using semantic similarity measure[9]. If there exists a previous query that is semantically related to the current query, it can be either suggested to the user, and internally used by the search engine to modify the original query[5].

III. Proposed Methodology

We proposed an automatic method to estimate the semantic similarity between words or entities using web search engine. Web search engine provide an efficient interface to this vast information. The figure1. illustrate the proposed system architecture to compute the semantic similarity between two words.

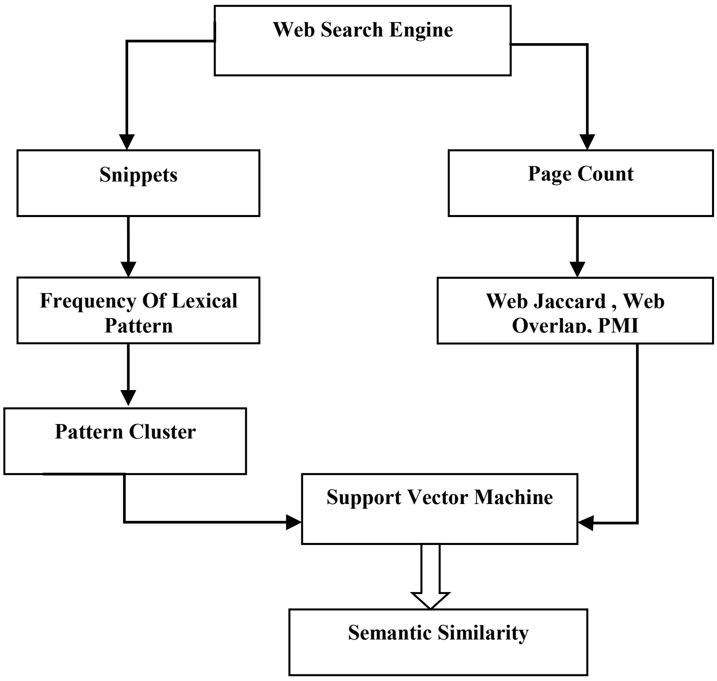


Fig. 1. Architecture of Proposed System

We query a web search engine and retrieve page counts for two words and for the conjunctive. The page count provides the similarity scores, web jaccard, web overlap, web PMI, consider the global co-occurrence of two words on the world web and the snippets represents the local content of two words on the web. Consequently we find the frequency of numerous lexical syntactic relationship between two words can convey accurately for the purpose a sequential pattern clustering algorithm are used to define various features. That representation the relation between two words using the feature representation of word pair can train by support vector machine.

A. Page Count

Page count of the query is an estimate of the number of pages that contain the query words. In general, page count may be necessarily equal to the word frequency because the queried word might appear many times on one page. Page count for the query P and Q can be Consider as a global measure of co-occurrence of words P and Q. Despite its simplicity, using page count alone as a measure of co-occurrence of two words presents several drawbacks. First, page count analysis ignores the position of the word in a page. Therefore, even though two words appear in a page, they might not be actually related. The page count of a polysemous word (a word with multiple senses) might contain a combination of all its senses.

B. Web Jaccard Module

The Web Jaccard coefficient between words (or multi word phrases) P and Q, Web Jaccard (P, Q)denotes the conjunction query P and Q. The scale and noise in web data, it is possible that two words may appear on some pages even though they are not related. To reduce the adverse effects attributable to such co-occurrences, we set the Web Jaccard coefficient to zero if the page count for the query (P and Q) is less than a threshold. information retrieval can be simply and rapidly with the use of search engine. This allows users to specify the search criteria as well as specific keywords to obtain the required result. Additionally, an index of search engines has to be updated on most recent information as it is constantly changed the time. Particularly, information retrieval results as documents are typically too extensive, which affect on accessibility of the required results for searchers.

Web Jaccard(P,Q)

$$= \begin{cases} 0, & \mathit{if}\ H(P \cap Q) \leq c, \\ \dfrac{H(P \cap Q)}{H(P) + H(Q) - H(P \cap Q)}, & \mathit{Otherwise} \end{cases} \qquad (1)$$

The equation(1) used to represents the notation H(P) denote the page count for the query P in a search engine. The Web Jaccard coefficient between words (multi-word phrases) P and Q, Web Jaccard (P,Q), is defined as, Therein, P∩Q denotes the conjunction query P AND Q. The scale and noise in Web data, it is possible that two words may appear on some pages even though they are not related. To reduce the adverse effects attributable to such co-occurrences, we set the Web Jaccard coefficient to zero the page count for the query P∩Q is less than a threshold c^2.

C. Web Overlap Module

Web Overlap is a natural modification to the Overlap coefficient. If user uses the web overlap module, it can produce meaning of the word. It helps the user to find correct meaning of the word or entities. measure the overlap across three major web search engine on the first results web overlap (i.e. share the same results) and the difference across a wide range of user defined search terms; determine the difference in the first page of search results and their rankings (each web search engine view of the most relevant content) across single-source web search engines, including both sponsored and non-sponsored result; and measure the degree to which a meta-search web engine, such as Dogpile.com, provides searcher with the most highly ranked search results from three major single source web search engines.

$$= \begin{cases} 0, & if\ H(P \cap Q) \le c, \\ \frac{H(P \cap Q)}{min(H(P)H(Q))}, & Otherwise \end{cases} \qquad (2)$$

Similarly, we define Web Overlap, this equation(2) Web Overlap(P,Q), as Web Overlap is a natural modification to the Overlap (Simpson) coefficient.

D. Web Point wise Mutual Information Module

The Point wise mutual information is a measure that is motivated by information theory; it is intended to reflect the dependence between two probabilistic events. To define Web PMI as a variant form of point wise mutual information using page counts. If user uses the web Point wise Mutual Information Module it can produce description of the word. By designing a new co-occurrence based word association measure by incorporating the concept of significant co-occurrence in the popular the word association measure to the Point wise Mutual Information . By extensive experiments with a large number of publicly available data sets we show that the newly introduced measure performs better than other co-occurrence based measures and despite the resource light, compares well with the best known resource-heavy distributed similarity and knowledge based word association measure.

$$= \begin{cases} 0, & if\ H(P \cap Q) \le c, \\ \frac{H(P \cap Q)}{H(P) + H(Q) - H(P \cap Q)}, & Otherwise \end{cases} \qquad (3)$$

We define Web PM Iin the equation(3) as a variant form of point wise mutual information using page counts. Here, N is the number of documents indexed by the search engine. Probabilities in are estimated according to the maximum likelihood principle. The calculate PMI accurately using, we must know N, the number of documents indexed by the search engine. The number of documents indexed by a search engine is an interesting task itself, it is beyond the scope of this work. We set N = 1010 according to the number of indexed ages reported by Google.

D. Lexical Pattern Extraction

In this module, Word in Page are extracted. It uses count-based co-occurrence measures. Lexical Pattern Clustering the problematic if one or both words are polysemous, when page counts are unreliable. On the other hand, the snippets returned by the search engine for the conjunctive query of two words provide useful clues related to the semantic relation that exist between the words. A snippet contains a window of text selected from a document that includes the queried word. Snippets are useful for search because most of the time, a user can read the snippet and decide whether the particular search result is relevant, without even opening the url. Using snippets as contexts is also computationally efficient because it obviate the need to download the source documents from the web, which can be time consuming if a document is large.

E. Lexical Pattern Clustering

Typically, the semantic relation can be expressed sing more than one pattern. For example, consider the two distinct pattern, X is a Y, and X is a large Y. Both these patterns indicate that there exists and is-a relation between X and Y. Identifying the different patterns that express the semantic relation enables us to represent the relation between two words accurately. According to the distributed hypothesis, words that occur in the same context have similar meanings. The distributed hypothesis has been used in various related tasks, such as identifying related words, and extracting paraphrases. If the consider to the word pairs that satisfy (i.e., co-occur with) a particular lexical pattern as the context of that lexical pair, then from the distributed hypothesis, it follows that the lexical patterns which are similarly distributed the word pairs must be semantically similar sequential pattern clustering algorithm

```
        Input: patterns $ʌ= \{a_1,...,a_n\}$, threshold
        Output: clusters C
1: SORT(ʌ)
2: C←{}
3: for pattern $a_i$ € ʌdo
4:   max ←−∞
5:   c*←null
6:   for cluster $c_j$ € C do
7:     sim ←cosine($a_i$,$c_j$)
8:     if sim > max then
9:       max ← sim
10:      c*← $c_j$
11:    end if
12:  end for
13:  if max > then
14:    c*← c* $a_i$
15:  else
16:    C ← C {$a_i$}
17:  end if
18: end for
19: return C
```

given a set ʌ of patterns and a clustering similarity threshold algorithm 1 returns clusters (of patterns)that express similar semantic relation. first, in algorithm1, the function sort sorts the patterns into descending order of their total occurrences in all word pairs. the occurrence $\mu(a)$ of a pattern a is the sum of frequencies over all word pairs. the most common patterns appear at the beginning in ʌ, whereas rare patterns (i.e., patterns that occur with only few word pairs) get shifted to the end. we initialize the set of clusters, c, to the empty set. the outer for-loop (starting at line 3), repeated takes a pattern a_i from the ordered set ʌ, and in the inner for-loop (starting at line 6), finds the cluster, c*(€ c) that is most similar to a_i. first, we represent a cluster by the centroid of all word pair frequency vector corresponding to the patterns in that cluster to compute the similarity between a pattern and a cluster. we consist the cosine similarity between the cluster centroid (c_j), and the word pair frequency vector of the pattern (a_i). the similarity between a pattern a_i, and its most similar cluster, c*, is greater than the threshold θ we append a_i to c* (line 14). we use the operator to denote the vector addition between c* and a_i. then we form a new cluster faig and append it to the set of clusters, c, if a_i is not similar to any of the existing clusters beyond the threshold θ.

F. Accurarcy Vs Number of Snippets

The correlation with the Miller-Charle ratings for different numbers of snippets to investigate the effect of the number of snippets used to extract pattern upon the semantic similarity measure. The experimental results are presented. It is apparent that overall the correlation coefficient improve with the number of snippets used for extracting pattern. The probability of finding better pattern increases with the number of processed snippets. That fact enables us to represent each pair of words with the rich feature vector, resulting in better performance.

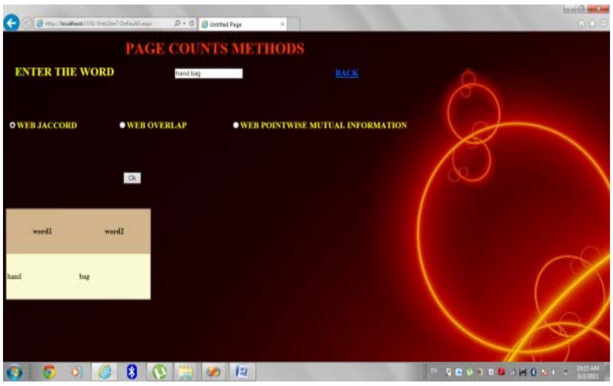
G. Measuring Semantic Similarity

We defined four co-occurrence measures using page count. We showed how to extract clusters of lexical patterns from snippets to represent numerous semantic relation that exist between two words. In this module, we describe a machine learning approach to combine both page count-based co-occurrence measures, and snippets-based lexical pattern clusters the construct a robust semantic similarity measure.
Abbreviations and Acronyms (Heading 2)

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE and SI do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

IV. EXPERIMENTAL RESULTS

The experimental result include semantic similarity between given two words by using SVM and page counts and text snippets retrieved from search engine for given words. Enter a key word in given text box to search in the search engine. For example, "Google" and "opera" are the word to search in web search engine. When click on search button, it display the page count and snippets as result. For example, the page counts and snippets for "Google" and "opera". The enter two words to measure similarity between them. the measurement ranges from 0 to 1. For given words "Google" and "opera" the semantic similarity is 0.8. For various words, we can measure semantic similarity between them. The result is close to 1 they are semantically closed and it is close to 0 when they are not closed semantically. The output will be shown in form of tables as follows. If the user is not satisfied with the displayed information then the page count method gives the more details about user input text.

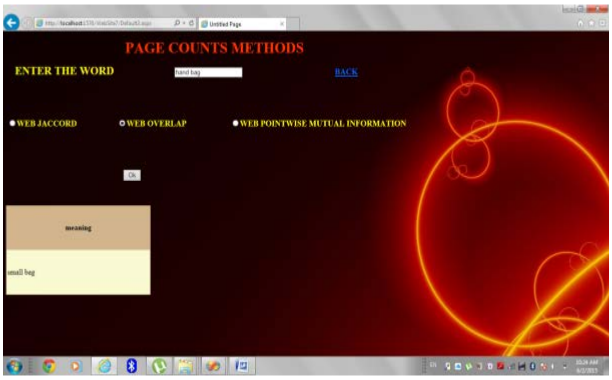


ENTER THE WORD hand bag

◉ WEB JACCORD ○ WEB OVERLAP ○ WEB POINTWISE MUTUAL INFORMATION

word1	word2
hand	bag

Fig. 2 Web Jaccard



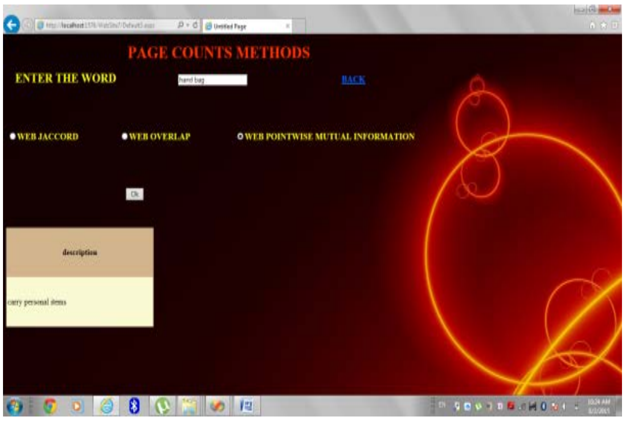
ENTER THE WORD hand bag

○ WEB JACCORD ◉ WEB OVERLAP ○ WEB POINTWISE MUTUAL INFORMATION

meaning
small bag

Fig. 3 Web Overlap

The represents the web search engine contain the two types, one is snippets and another one is page count. The web search function is to search the word the entered word is moved to the page count which in turn page count will process the given word in the server and provide the number of times the word is present. The searched word is then moved to snippets, the snippets will process and show how many relationship it have.



ENTER THE WORD hand bag

○ WEB JACCORD ○ WEB OVERLAP ◉ WEB POINTWISE MUTUAL INFORMATION

description
carry personal items

Fig. 4 Web Point wise Mutual Information

The output of snippets is given to the frequency of lexical pattern as input, the frequency of lexical pattern will analyze the similarities. The resulting output will be transfer to pattern cluster, it will cluster those words and find a suitable word that is satisfied to the user and will be given to support vector machine the page count consists of three modules 1. web jaccard 2. web overlap 3. point wise mutual information. The pattern cluster output and page count output is sent to the support vector machine . The support vector machine will provide the semantic similarity of the given word.

V. Conclusion

A semantic similarity measure using page count and snippets retrieved from a web search engine for two words co-occurrence measures were computed using page counts. Lexical pattern extraction algorithm to extract numerous semantic relation that exist between the two words. Moreover, a sequential pattern lexical patterns that describe the same semantic relation. The page counts-based co-occurrence measures and lexical pattern clusters were used to define features for a word pair. The two-class SVM was trained using those features extracted for synonymous and no synonymous word pairs selected from Word Net synsets. Showed that the proposed method outperform various baselines as well as previously proposed web-based semantic similarity measure, achieving a high correlation with human ratings. Moreover, the proposed method improved in a community mining, thereby underlining its usefulness in real-world tasks that include named entities not adequately covered by manually created resources.

References

[1] Danuska Bollegala,Yutaka Matsuo, and MitsuruMitsuru Ishizuka, “A Web Search Engine-based Approach to Measure Semantic Similarity between Words,” Proc. of ACM SIGIR Conference, 2012.

[2] E. Agichtein, E. Brill, S. Dumais, and R. Rango, “Improving Web Search Ranking by Incorporating User Behavior Information,” Proc. of ACM SIGIR Conference, 2006.

[3] R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” Proc. of ACM SIGMOD Conference, 2011.

[4] R. A. Baeza-Yates, C. A. Hurtado and M. Mendoza, “Query Recommendation using Query Logs in Search Engines,” EDBT Workshop, vol. 3268, pp. 588-596, 2004.

[5] D. Beeferman and A. Berger, “Agglomerative Clustering of a Search Engine Query Log,” Proc. of ACM SIGKDD Conference, 2013.

[6] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman and O. Frieder, “Hourly Analysis of a Very Large Topically Categorized Web Query Log,” Proc. of ACM SIGIR Conference, 2011.

[7] V.W. Chan, K.W. Leung, and D.L. Lee, “Clustering Search Engine Query Log Containing Noisy Clickthroughs,”Proc. of SAINT Conference, 2004.

[8] S. Chuang and L. Chien, “Automatic Query Taxonomy Generation for Information Retrieval Applications,” Online Information Review, vol. 27, Issue 4, pp. 243-255, 2003.

[9] H. Cui, J. Wen, J. Nie and W. Ma, “Query Expansion by Mining User Logs,” IEEE TKDE, vol. 15, Issue 4, pp. 829-839, 2003.

[10] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, “A study on similarity and relatedness using distributional and word net-based approaches,” in Proc. of NAACL-HLT’09,2009.

[11] G. Hirst, , and D. St-Onge, “Lexical chains as representations of context for the detection and correction of malapropisms.”Word Net: An Electronic Lexical Database, pp. 305–?32, 1998.