Semantic Similarity Using Web Search Engine

Meghana Raut[1], Nityaspandana Nalamari[2], Darshana Rane[3]

[1, 2, 3]Student, Bharati Vidyapeeth college of Engineering for women, Pune University, Katraj, Pune-043, India

Abstract: ***Measuring the semantic similarity between words is an important component in various tasks on the web such as relation extraction, document clustering, and automatic metadata extraction. Despite the usefulness of semantic similarity measures in these applications, accurately measuring semantic similarity between two words (or entities) is still difficult. We propose a method to estimate semantic similarity using page counts and text snippets retrieved from a web search engine for two words. Specifically, we define various word co-occurrence measures using page counts and integrate those with lexical patterns extracted from text snippets. To identify the numerous semantic relations that exist between two given words, we propose a pattern extraction algorithm and a pattern clustering algorithm. The optimal combination of page counts-based co-occurrence measures and lexical pattern clusters is obtained using support vector machines.***

Keywords: web mining, information retrieval, page counts, snippets

1. Introduction

Accurately measuring the semantic similarity between words is an important problem in web mining and information retrieval. Web mining applications such as community extraction, relation detection, require the ability to accurately measure the semantic similarity between concepts or entities. In information retrieval, one of the main problems is to retrieve a set of documents that is semantically related to a given user query. Efficient estimation of semantic similarity between words is critical for various natural language processing tasks such as word sense disambiguation (WSD), textual entailment, and automatic text summarization.

We propose an automatic method to estimate the semantic similarity between words or entities using web search engines. Because of the vastly numerous documents and the high growth rate of the web, it is time consuming to analyze each document separately. Web search engines provide an efficient interface to this vast information. Page counts and snippets are two useful information sources provided by most web search engines. Page count of a query is an estimate of the number of pages that contain the query words. Snippets, a brief window of text extracted by a search engine around the query term in a document, provide useful information regarding the local context of the query term. Semantic similarity measures defined over snippets, have been used in query expansion [3], personal name disambiguation [4], and community mining [5]. Processing snippets is also efficient because it obviates the trouble of downloading web pages, which might be time consuming depending on the size of the pages. The proposed method considers both page counts and lexical syntactic patterns extracted from snippets to produce accuracy in the results obtained after search for a particular query made in the search engine.

2. Problem Statement

To propose a method to evaluate semantic similarity using page counts and text snippets retrieved from a web search engine for two words.

3. Related Work

Sahami and Heilman [3] measured semantic similarity between two queries using snippets returned for those queries by a search engine. For each query, they collect snippets from a search engine and represent each snippet as a TF-IDF-weighted term vector. Each vector is L2 normalized and the centroid of the set of vectors is computed. Semantic similarity between two queries is then defined as the inner product between the corresponding centroid vectors. Given taxonomy of words, a straightforward method to calculate similarity between two words is to find the length of the shortest path connecting the two words in the taxonomy [6].But they did not compare their similarity measure with taxonomy-based similarity measures.

Cilibrasi and Vitanyi [2] proposed a distance metric between words using only page counts retrieved from a web search engine. The proposed metric is named Normalized Google Distance (NGD). NGD is based on normalized information distance, which is defined using Kolmogorov complexity. Because NGD does not take into account the context in which the words co-occur, they consider only page counts.

4. Proposed System

We first develop a search engine with some basic functionality using Apache Lucene. Web search engine provide an efficient interface to the vast information. Then we propose an automatic method to estimate the semantic similarity between words or entities using this web search engine. Page counts and snippets are two useful information sources provided by the web search engine. Page count of a query is an estimate of the number of pages that contain the query words. In general, page count may not necessarily be equal to the word frequency because the queried word might appear many times on one page. We present an automatically extracted lexical syntactic patterns-based approach to compute the semantic similarity between words or entities using text snippets retrieved from a web search engine. Then we combine this both page count-based co-occurrences measures and lexical pattern clusters to train SVM that distinguishes between synonymous and non-synonymous word pairs.

5. Approach

First, we construct a web search engine using Apache Lucene tool. Lucene is a open source information retrieval software library. It is utilized for implementing internet search engines. It performs the function of indexing and search. Next, after obtaining the results from lucene we compute the co-occurrences of words in a particular file using page counts [1]. This method uses four co-occurrence measures such as WebJaccard, WebOverlap, WebDice, WebPMI.

The WebJaccard coefficient between words (or multiword phrases) P and Q,

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

P∩Q denotes the conjunction query P AND Q. H(P) indicates the page count for P. H(Q) indicates page count of Q. Given the scale and noise in web data, it is possible that two words may appear on some pages even though they are not related. In order to reduce the adverse effects attributable to such co-occurrences, we set the WebJaccard coefficient to zero if the page count for the query P∩Q is less than a threshold c. Where c=5. Similarly, the rest of the co-efficient are calculated.

$$\text{WebOverlap}(P,Q) = \begin{cases} 0, & \text{if } H(P \cap Q) \leq c, \\ \dfrac{H(P \cap Q)}{\text{Min}(H(P),H(Q))} & \text{otherwise} \end{cases}$$

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

$$\text{WebPMI}(P,Q) = \begin{cases} 0, & \text{if } H(P \cap Q) \leq c, \\ \text{Log}_2 \left(\dfrac{\frac{H(P \cap Q)}{N}}{\frac{H(P)}{N}\frac{H(Q)}{N}} \right) & \text{otherwise} \end{cases}$$

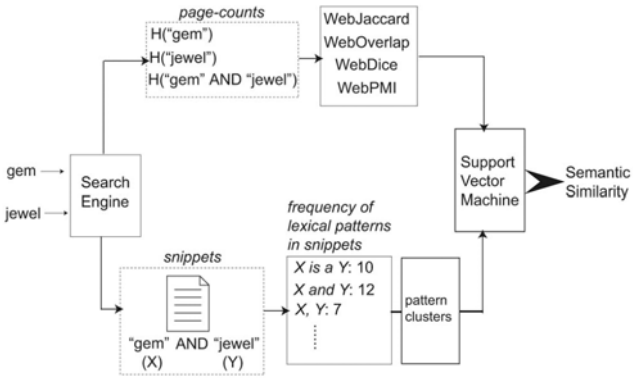
Where PMI= Pointwise Mutual Information

After, calculating the page count of the two words entered as a query, we check the local context of those two words using snippets. In order to identify the patterns that exists between the two words that is the relation of the two words in the snippets we use lexical pattern extraction. For a snippet α, retrieved for a word pair (P,Q) first, we replace the two words P and Q, respectively, with two variables X and Y. We replace all numeric values by D, a marker for digits. Next, we generate all subsequences of words from α that satisfy all of the following conditions:

a) A subsequence must contain exactly one occurrence of each X and Y.
b) The maximum length of a subsequence is L words. A subsequence is allowed to skip one or more words. However, we do not skip more than g number of words consecutively.
c) Moreover, the total number of words skipped in a subsequence should not exceed G. We expand all negation contractions in a context. For example, “didn’t” is expanded to “did not”.
d) We do not skip the word ‘not’ when generating subsequences. For example, this condition ensures that from the snippet X is not a Y, we do not produce the subsequence X is a Y.
e) Finally, we count the frequency of all generated subsequences and only use subsequences that occur more than T times as lexical patterns.
f) The values L=5, g=2, G=4, T=5 are set experimentally.

Now, we implement the sequential pattern clustering algorithm to identify different lexical pattern that describe the same semantic relation. A feature f is defined for word pair using both page count-based co-occurrences measures and lexical pattern clusters.[1] We then train a two class Support Vector Machine to classify synonymous and non-synonymous word pairs. The support vector machine then combines both the results of page count co-occurrence and lexical pattern extracted to identify accurate semantic relations and provide the results accordingly.

6. Figures and Tables



The above image provides a basic idea of the system being developed.

7. Assumption and Dependencies

1. We are using Word Net online dictionary to extract word pairs.
2. We assume that all patterns in a cluster represent a particular semantic relation.
3. We are using Apache Lucene to construct web search engine.
4. We are going to use directly available libraries to train SVM.

8. Conclusion

Semantic similarity measure is obtained using both page counts and snippets retrieved from a web search engine for two words. Four word co-occurrence measures were

computed using page counts. We proposed a lexical pattern extraction algorithm to extract numerous semantic relations that exist between two words. Moreover, a sequential pattern clustering algorithm was proposed to identify different lexical patterns that describe the same semantic relation. Both page counts-based co-occurrence measures and lexical pattern clusters were used to define features for a word pair. A support vector machine was trained using those features extracted for synonymous and nonsynonymous word pairs selected from WordNet synsets. Due to the training provided to the support vector machine the similarity measures between two words can be calculated automatically and almost exactly. Therefore, the number of results thus produced by the search engine would be reduced considerably that is irrelevant options are removed and only the essential results are finally displayed to the user. Thus, this approach is better than any other technique for providing similarity results.

References

[1] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring Semantic Similarity between Words Using Web Search Engines," IEEE transactions on knowledge and data engineering, vol. 23, no. 7, July 2011.
[2] R. Cilibrasi and P. Vitanyi, "The Google Similarity Distance," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 3, pp. 370-383, Mar.2007.
[3] M. Sahami and T. Heilman, "A Web-Based Kernel Function for Measuring the Similarity of Short Text Snippets," Proc. 15th Int'l World Wide Web Conf., 2006.
[4] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Disambiguating Personal Names on the Web Using Automatically Extracted Key Phrases," Proc. 17th European Conf. Artificial Intelligence, pp. 553-557, 2006.
[5] H. Chen, M. Lin, and Y. Wei, "Novel Association Measures Using Web Search with Double Checking," Proc. 21st Int'l Conf. Computational Linguistics and 44th Ann. Meeting of the Assoc. for Computational Linguistics (COLING/ACL '06), pp. 1009-1016, 2006.
[6] R. Rada, H. Mili, E. Bichnell, and M. Blettner, "Development and Application of a Metric on Semantic Nets," IEEE Trans. Systems, Man and Cybernetics, vol. 19, no. 1, pp. 17-30, Jan./Feb. 1989.
[7] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring Semantic Similarity between Words Using Web Search Engines," Proc. Int'l Conf. World Wide Web (WWW '07), pp. 757-766, 2007.

Author Profile

Meghana Raut is a final year computer engineering student of Bharati Vidyapeeth College of engineering for women. The college is affliated to Pune University.

Nityaspandana Nalamari is a final year computer engineering student of Bharati Vidyapeeth College of engineering for women. The college is affliated to Pune University.

Darshana Rane is a final year computer engineering student of Bharati Vidyapeeth College of engineering for women. The college is affiliated to Pune University.