International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Web Mining for Semantic Similarity between Words

Shirajuddin A. Shaikh[1], Anis Fatima N. Mulla[2]

[1, 2]Annasaheb Dange College of Engineering and Technology Ashta Sangli Maharashtra India

Abstract: ***Semantic similarity is a context dependent and dynamic phenomenon. A train is semantically similar to a horse if the context is moving objects. On the otherhand if the context is living beings the same objects may not have any similarity. Finding semantic similarity has increasingly become important in many applications such as community mining, data clustering, relation extraction, query expansion and many natural language processing projects. The paper addresses the general framework for finding semantic similarity between words and its utility in the state of art web applications. It presents some empirical results using page count and snippet based methods on the web.***

Keywords: semantic similarity, web mining, text mining, computational linguistics.

1. Introduction

Similarity is a fundamental concept in theories of knowledge discovery. It acts as an organizing principle by which individuals classify objects, and make generalizations (Goldstone, 1994). The usefulness of semantic similarity measures in these applications, accurately measuring semantic similarity between two words (or entities) remains a challenging task. In the literature of data mining there are several similarity measures viz web jaccard, web overlap, web dice and web pmi to find similarity between objects based on the data type of objects. However the point here is, finding similarity becomes exceedingly challenging if the context of the object is considered along with its data type.

In this paper a general framework for finding context dependent similarity between objects is presented and applied to find similarity between words in the context of their occurrences in the web pages. This context dependent similarity between words is referred as semantic similarity between words. Suppose **P** & **Q** are two words of our interest, then we would be interested to calculate the semantic similarity between **P** & **Q**. In the context of web search we explore the following methods to capture semantic similarity between words.

1. Page count
2. Snippets

Let,

N1 be the number pages in the web containing the word **P** when it is searched as a single word.

N2 be the number pages in the web containing the word **Q** when it is searched as a single word.

N12 be the number pages in the web containing the words **P** & **Q** when it is searched for combination of words.

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. In general, page count may not necessarily be 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 considered as a global measure of cooccurrence of words **P** and **Q**. For example, the page count of the query apple AND computer in Google is 288,000,000, whereas the same for banana AND computer is only 3,590,000. The more than 80 times more numerous page counts for apple AND computer indicate that apple is more semantically similar to computer than is banana.

Given two words **P** and **Q**, we model the problem of measuring the semantic similarity between **P** and **Q**, as a one of constructing a function sim (P,Q)that returns a value in range [0,1]. If **P** and **Q** are highly similar (e.g., synonyms), we expect sim (**P**,**Q**) to be closer to 1. On the other hand, if **P** and **Q** are not semantically similar, then we expect sim (**P,Q**) to be closer to 0. We define numerous features that express the similarity between **P** and **Q** using page counts and snippets retrieved from a web search engine for the two words.

2. Related Work

In order to find the similarity between two words the most straightforward method is to find the length of the shortest distance between them by classifying them depending on their classification. But if a word has multiple meanings which mean having multiple paths then the shortest path between the senses of the word is considered for calculating similarity. A problem that is frequently acknowledged with this approach is that it relies on the notion that all links in the taxonomy represent a uniform distance. The similarity between two words can be also calculated using the information content in the words. It is calculated as the maximum of the information content of all concepts C that include both **C1** and **C2**. This was proposed by Resnick [2] and the class that was used as the taxonomy was WordNet.

Li[3] proposed a similarity measure that uses shortest path length, depth and local Density in taxonomy. The dataset that

was used was of Miller and Charles benchmark dataset. The Pearson correlation coefficient of 0:8914 was obtained, but they did not evaluate their method in terms of similarities among named entities.

Lin [4] defined the similarity between two concepts as the information that is in common to both concepts and the information contained in each individual concept.

The normalized Google Distance (NGD) gained a lot of popularity, and was proposed by Cilibrasi and Vitanyi [5]. It took into consideration the page counts returned by the Google Search Engine and the formula is given by:

$$NGD(P,Q) = \frac{\max\{\log H(P), \log H(Q)\} - \log H(P,Q)}{\log N - \min\{\log H(P), \log H(Q)\}}. \quad ..(1)$$

Here, **P** and **Q** are the two words between which distance NGD(**P**,Q) is to be computed, H(**P**) denotes the page-counts for the word **P**, and H(**P**,**Q**) is the pagecounts for the query **P** AND **Q**. NGD is based on normalized information distance [6], which is defined using Kolmogorov complexity.

The drawback of this method was that the context in which the words appear are not taken into consideration. In order to remove this drawback, the semantic similarity is taken into consideration by Sahami. Sahami et al., [7] 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.

Semantic similarity between two queries is then defined as the inner product between the corresponding centroid vectors.The drawbacks was that the taxonomy or the word classification was not taken into consideration.

Chen et al., [5] proposed a double-checking model using text snippets returned by a Web search engine to compute semantic similarity between words. For two words **P** and **Q,** they collect snippets for each word from a Web search engine. Then they count the occurrences of word **P** in the snippets for word **Q** and the occurrences of word **Q** in the snippets for word **P**. The Co-occurrence Double-Checking (CODC) measure is defined as,

$$\mathrm{CODC}(P,Q) = \begin{cases} 0, & \text{if } f(P@Q) = 0, \\ \exp\left(\log\left[\frac{f(P@Q}{H(P)} \times \frac{f(Q@P)}{H(Q)}\right]^{\alpha}\right), & \text{otherwise.} \end{cases} \quad ..(2)$$

Here, f(**P@Q**) denotes the number of occurrences of **P** in the top-ranking snippets for the query **Q** in Google, H(**P**) is the page count for query **P**, and _ is a constant in this model, which is experimentally set to the value 0:15. But this method also was not error free as the web does not rank the pages only on the basis of snippets. It also takes into consideration the date of publication, the link structure etc. Semantic similarity measures have been used in various applications in natural language processing such as word-sense disambiguation, language modeling, synonym extraction and automatic thesauri extraction. Semantic similarity measures are important in many Web-related tasks.

3. Proposed Method

The proposed method is aimed at overriding most of the drawbacks in the algorithms mentioned above. The proposed method tries to use the page counts returned by the search engines as well as the lexical pattern similarity between the snippets returned by the web search engine.

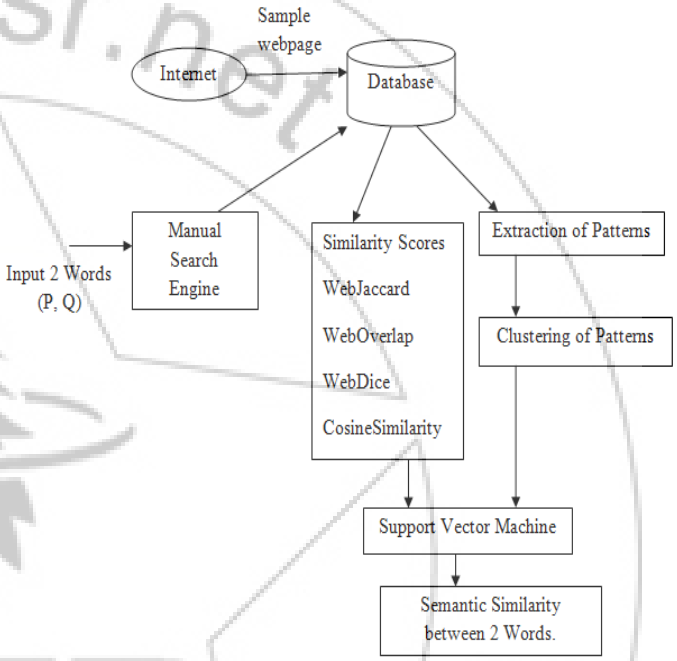

Figure 1: Outline of the proposed method

Four popular similarity scores using page counts are used here. Page counts-based similarity scores consider the global co-occurrences of two words on the web. They are: WebJaccard, WebOvelap, WebDice and WebPMI. After calculating the four parameters, the snippets from the web are obtained. The lexical patterns are extracted and clusters are formed. After the clusters, a feature is extracted from each of the clusters. After getting the (n+1) features, where n is the no. of clusters a 2 class SVM is implemented so as to give the similarity measure. The similarity measure is close to 0% if the words are dissimilar and the similarity measure is close to 100% if the words are similar.

4. Method

4.1 Page counts based Similarity Scores

Page counts for the query **P AND Q** can be considered as an approximation of co-occurrence of two words (or multi-word phrases) **P** and **Q** on the Web. We compute four popular co-occurrence measures; Jaccard, Overlap (Simpson), Dice, and Pointwise mutual information (PMI), to compute semantic similarity using page counts. For the remainder of this paper we use the notation H(**P**) to denote the page counts for the

query **P** in a search engine. The WebJaccard coefficient between words (or multi-word phrases) **P** and **Q**, WebJaccard (**P**,**Q**), is defined as

$$\text{WebJaccard}(P,Q) = \begin{cases} 0, & \text{if } H(P \cap Q) \leq c, \\ \dfrac{H(P \cap Q)}{H(P) + H(Q) - H(P \cap Q)}, & \text{otherwise.} \end{cases} \quad ..(3)$$

Therein, $P \cap Q$ **Q** denotes the conjunction query **P** AND **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 $P \cap Q$ is less than a threshold c2.

Similarly, we define WebOverlap: WebOverlap (**P**,**Q**), as,

$$\text{WebOverlap}(P,Q) = \begin{cases} 0, & \text{if } H(P \cap Q) \leq c, \\ \dfrac{H(P \cap Q)}{\min(H(P), H(Q))}, & \text{otherwise.} \end{cases} \quad ..(4)$$

WebOverlap is a natural modification to the Overlap (Simpson) coefficient. We define the WebDice coefficient as a variant of the Dice coefficient.

WebDice (P,Q) is defined as,

$$\text{WebDice}(P,Q) = \begin{cases} 0, & \text{if } H(P \cap Q) \leq c, \\ \dfrac{2H(P \cap Q)}{H(P) + H(Q)}, & \text{otherwise.} \end{cases} \quad ..(5)$$

we set c = 5 in our experiments

Pointwise mutual information (PMI) [20] is a measure that is motivated by information theory; it is intended to reflect the dependence between two probabilistic events. We define WebPMI as a variant form of pointwise mutual information using page counts as,

$$\text{WebPMI}(P,Q) = \begin{cases} 0, & \text{if } H(P \cap Q) \leq c, \\ \log_2 \left(\dfrac{\frac{H(P \cap Q)}{N}}{\frac{H(P)}{N} \frac{H(Q)}{N}} \right), & \text{otherwise.} \end{cases} \quad ..(6)$$

4.2 Lexical Pattern Extraction

A snippet contains a window of text selected from a document that includes the queried words. Snippets are useful for search because, most of the time, a user can read the snippet and decide whether a particular search result is relevant, without even opening the url. Using snippets as contexts is also computationally efficient because it obviates the need to download the source documents from the Web, which can be time consuming if a document is large.

"**Cricket** is a **sport** played between two teams, each with eleven players."

The phrase indicates a semantic relationship between **cricket** and **sport**. Many such phrases indicate semantic relationships. In the example given above, words indicating the semantic relation between cricket and sport appear between the query words. Replacing the query words by variables **X** and **Y** we can form the pattern **X** is a **Y** from the example given above.

4.3 Clustering

Typically, a semantic relation can be expressed using more than one pattern. For example, consider the two distinct patterns, **X** is a **Y**, and **X** is a large **Y**. Both these patterns indicate that there exists an is-a relation between **X** and **Y**. Identifying the different patterns that express the same semantic relation enables us to represent the relation between two words accurately. The frequency of all the patterns that are collected, is calculated over each word pair by the formula: The total occurrence μ(a) of a pattern a is the sum of frequencies over all word pairs, and is given by,

$$\mu(a) = \sum_i f(Pi, Qi, a) \quad ..(7)$$

These patterns are then sorted according to their frequencies, such that the most frequent pattern is at the top and the rarest is at the bottom. The clusters (n)'s centroids are formed randomly and the cosine similarity between the patterns and the centroids is calculated so as to form clusters for more common relations first. This enables us to separate rare patterns which are likely to be outliers from attaching to otherwise clean clusters.

4.4 Measuring Semantic Similarity

Now that 4 features of each word pair calculated from the page counts similarity scores, and calculate a feature vector for each word pair, which is of N+ 4 lengths. Compute the value of the j-th feature in the feature vector for a word pair (P,Q) as follows,

$$\sum_{ai \in cj} wij\, f(P,Q,ai) \quad ..(8)$$

The value of the j-th feature of the feature vector fPQ representing a word pair (P,Q) can be seen as the weighted sum of all patterns in cluster cj that co-occur with words P and Q. After this we get the (N+4) feature vector for each word pair, which should be then submitted to the SVM class for training.

4.5 SVM

The SVM module consists of the training and the testing module.

5. Training Database

The dataset used for the training is taken from the WordNet. Numerous words with their synonyms are taken and for the unrelated words, these synonyms are randomly swapped. The

entire set is then supplied to the SVM. After training, the train model is saved, which is then used while testing.

6. Database and Result

wordpair	webjaccard	weboverlap	webdice	webpmi
journey voyage	0.09649122807017543	0.2357142857142857	0.176	2.5287635480034094
gem jewel	0.19106699751861042	0.393526405451448	0.32083333333333336	3.8315588081722463
boy lad	0.09252265861027191	1.4583333333333333	0.16937435188385758	3.2868488084544065
furnace stove	0.2056074766355514	0.4247104247104247	0.34108527131782945	4.700740399598276
coast shore	0.41464325737766156	1.1721224920802535	0.5862160021124901	3.7201823349352785
magician wizard	0.019793605142953814	0.09227129337539432	0.038818845388818845	1.355047498105293
bird cock	0.025810721376571807	0.05064935064935065	0.05032258064516129	0.7721903879003984
bird crane	0.11476725521669343	0.6111111111111112	0.2059035277177826	3.275446176356895
implement tool	0.16533455126741264	0.9366106080206986	0.2837546541250245	3.074114987771987
car journey	0.21428571428571427	1.1808510638297873	0.35294117647058826	2.401161863562561
midday noon	0.09757705620570298	0.7141133896260555	0.17780447514641841	4.808024758368447
food fruit	0.4746494066882416	2.4858757062146895	0.6437454279444038	3.0392567802090853
monk oracle	0.0213903743315508	0.07168458781362007	0.041884816753926704	2.36125288065989
brother monk	0.05058823529411765	0.46236559139784944	0.09630459126539753	2.958302071174498
brother lad	0.06625097427903351	0.5059523809523809	0.12426900584795321	3.048388725721249
food rooster	0.020204581959590837	1.4226190476190477	0.039608882996354	2.4811313583988595
monk slave	0.03268332875583631	0.127956989247311 83	0.06329787234042553	2.7128051856231 09
lad wizard	0.03158566414973944	0.08839285714285715	0.06123711340206186	2.634912396378452
cord smile	0.016570523696414433	0.0881203007518797	0.03260083449235049	1.1011154188862282
p?db=webmining&table=features&token=b5fd980a3a40cfa7...		78991597	0.037407407407407 41	0.3401781047770521

Figure 2: Database and Result Table

The scores like webjaccard, weboverlap, webdice are closer to 0 when the word pairs are dissimilar and the values are more when the words are similar.

7. Conclusion

The proposed a semantic similarity measures 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 cooccurrence measures and lexical pattern clusters were used to define features for a word pair. A two-class SVM was trained using those features extracted for synonymous and non-synonymous word pairs selected from WordNet synsets.

References

[1] D.Bollegala, Y.Matsuo and M.ishizuka, “A Web Search Engine-Based Approach to Measure Semantic Similarity between Words,” IEEE Trans. Knowledge and Data Eng, vol.23, no.7, pp. 977-990, july 2011.
[2] P. Resnik, “Using information content to evaluate semantic similarity in a taxonomy,” in Proc. of 14th International Joint Conferenceon Aritificial Intelligence, 1995.
[3] D. M. Y. Li, Zuhair A. Bandar, “An approch for measuring semantic similarity between words using multiple information sources,” IEEE Transactions on Knowledge and Data Engineering, vol. 15(4), pp. 871–882, 2003.
[4] D. Lin, “An information-theoretic definition of similarity,” in Proc. of the 15th ICML, 1998, pp. 296–304.
[5] R. Cilibrasi and P. Vitanyi, “The google similarity distance,” IEEE Transactions on Knowledge and Data Engineering, vol. 19, no.3, pp.370–383, 2007
[6] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi, “The similarity metric,” IEEE Transactions on Information Theory, vol. 50, no. 12, pp. 3250–3264, 2004.
[7] 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.
[8] D. Mclean, Y. Li, and Z.A. Bandar , “An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources,” IEEE Trans. Knowledge and Data Eng., vol. 15, no. 4, pp. 871-88,,2006.
[9] Z. B.Yossef and M. Gurevich, “Sampling from a Search Engine Index,” Proc. 15th Int World Wide Web Conf., 2006.
[10] R. Rada, H. Mili, E. Bichnell, and M. Blettner, “Development and Application of a Metric on Semantic Nets,” Trans. Systems, Man and Cybernetics, vol. 19, no. 1, pp.17-30, Jan / Feb. 1989.

Author Profile

Shirajuddin A. Shaikh received the B.E (IT) from Dr.J.J Magdum College of Engineering Jaysingpur Maharashtra India and currently is pursuing M.E. from Department of Computer Science and Engineering at Annasaheb Dange College of Engineering and Technology Ashta Sangli Maharashtra India.

Anis Fatima N. Mulla working as Assistant Professor in Department of Computer Science and Engineering at Annasaheb Dange College of Engineering and Technology Ashta Sangli Maharashtra India. She has completed her master's degree in Computer Science and Engineering.