Measuring Semantic Similarity between Words Using Web Documents

Sheetal A. Takale
Information Technology Dept.
Vidya Pratishthans's College Of
Engineering, Baramati India
sheetaltakale@rediffmail.com

Sushma S. Nandgaonkar
Computer Engineering Dept.
Vidya Pratishthans's College Of
Engineering, Baramati India
sushma.nandgaonkar@gmail.com

***Abstract*— Semantic similarity measures play an important role in the extraction of semantic relations. Semantic similarity measures are widely used in Natural Language Processing (NLP) and Information Retrieval (IR). The work proposed here uses web-based metrics to compute the semantic similarity between words or terms and also compares with the state-of-the-art. For a computer to decide the semantic similarity, it should understand the semantics of the words. Computer being a syntactic machine, it can not understand the semantics. So always an attempt is made to represent the semantics as syntax. There are various methods proposed to find the semantic similarity between words. Some of these methods have used the precompiled databases like WordNet, and Brown Corpus. Some are based on Web Search Engine. The approach presented here is altogether different from these methods. It makes use of snippets returned by the Wikipedia or any encyclopedia such as Britannica Encyclopedia. The snippets are preprocessed for stop word removal and stemming. For suffix removal an algorithm by M. F. Porter is referred. Luhn's Idea is used for extraction of significant words from the preprocessed snippets. Similarity measures proposed here are based on the five different association measures in Information retrieval, namely simple matching, Dice, Jaccard, Overlap, Cosine coefficient. Performance of these methods is evaluated using Miller and Charle's benchmark dataset. It gives higher correlation value of 0.80 than some of the existing methods**

Keywords – Semantic Similarity, Wikipedia, Web Search Engine, Natural Language Processing, Information Retrieval, Web Mining.

I. INTRODUCTION

Semantic similarity is a central concept that finds great importance in various fields such as artificial intelligence, natural language processing, cognitive science and psychology. Accurate measurement of semantic similarity between words is essential for various tasks such as, document clustering, information retrieval, and synonym extraction. For a machine to be able to decide the semantic similarity, intelligence is needed. It should be able to understand the semantics or meaning of the words. But a computer being a syntactic machine, semantics associated with the words or terms is to be represented as syntax. For this various approaches are proposed till now. Word semantic similarity approaches or metrics can be categorized as: (i) *Pre-compiled database based metrics*, i.e., metrics consulting only human-built knowledge resources, such as ontologies, (ii) *Co-occurrence based metrics using WWW*, i.e., metrics that assume that the semantic similarity between words or terms can be expressed by an association ratio which is a function of their co-occurrence (iii*) Context based metrics using WWW*, i.e., metrics that are fully text-based and understand and utilize the context or proximity of words or terms to compute semantic similarity.

Several Precompiled database based methods have been proposed in the literature that use, e.g., WordNet, for semantic similarity computation. WordNet is an on-line semantic dictionary—a lexical database, developed at Princeton by a group led by Miller. Edge counting methods consider the length of the paths that link the words, as well as the word positions in the taxonomic structure [4]. Information content methods compute similarity between words by combining taxonomic features that exist in the used resource, e.g., number of subsumed words, with frequencies computed over textual corpora [3]. Semantic similarity between words changes over time as new words are constantly being created and new meaning is also being assigned to the existing words. Also there can be a problem with person name detection and alias detection. One person may have multiple names to identify. So there are some problems with the precompiled databases. The new senses of words can not be immediately listed in any precompiled database. Maintaining an up-to-date taxonomy of all the new words and new usages of existing words is difficult and costly. A solution to this problem is : "*The Web can be regarded as a large-scale, dynamic corpus of text*". Danushka Bollegala [6] has proposed similarity measures using page count returned by the search engine for the given word pair. These similarity measures are modified four popular co-occurrence measures; Jaccard, Overlap, Dice, and PMI (point-wise mutual information). Page-count-based metrics use association ratios between words that are computed using their co-occurrence frequency in documents. The basic assumption of this approach is that high co-occurrence frequencies indicate high association ratios and high association ratios indicate a semantic relation between words.

Cilibrasi and Vitanyi [7] proposed a page-count-based similarity measure, called the Normalized Google Distance.

$$G(w_1, w_2) = \frac{\max\{A\} - \log|D|w_1, w_2)}{\log|D| - \min\{A\}} \qquad (1)$$

As the semantic similarity between two words increases, the distance computed by (1) decreases. This metric is considered to be a dissimilarity measure. The metric is also unbounded, ranging from 0 to ∞. J. Gracia [5], proposed a variation of Normalized Google Distance that defines a similarity measurement. This variation is typically referred to as "Google-based Semantic Relatedness":

$$G'(w_1, w_2) = e^{-2G(w_1, w_2)} \quad (2)$$

The next approach is using TF-IDF representation to represent semantics of a word. Here Term Frequency (TF) is the ratio of number of occurrences of the considered term (t_i) in document d_j, and the total number of occurrences of all terms in document d_j.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (3)$$

Inverse Document Frequency (IDF) is the ratio of total number of documents and the number of documents having the term t_i .

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|} \quad (4)$$

TF-IDF is

$$(tf - idf)_{i,j} = tf_{i,j} \times idf \quad (5)$$

Elias Iosif [8] proposed text-based or context based similarity metrics. The basic assumption behind these metrics is that "*similarity of context implies similarity of meaning"*, i.e., words that appear in similar lexical environment (left and right contexts) have a close semantic relation. For each occurrence of a word w a left and right context of size K is considered. i.e. $[t_{K,L} ... t_{2,L} t_{1,L}] w [t_{1,R} t_{2,R} ... t_{K,R}]$ where,

$t_{i,L}$ and $t_{i,R}$ represent the i^{th} word to the left and to the right of w respectively. Each word is represented as a feature vector as $\mathrm{F}_{w,K} = (\mathrm{v}_{w,1}, \mathrm{v}_{w,2}, ..., \mathrm{v}_{w,N})$. There are various feature weighting schemes for computing the value of $\mathrm{v}_{w,i}$, some of them are :

Scheme	**Acronym**
Binary	B
Term Frequency	TF
Add-one TF	TF1
Log of TF	LTF
Add-one LTF	LTF1
TF-Inverse Document Freq.	TFIDF
Log of TFIDF	LTFIDF
Add-one LTFIDF	LTFIDF1

This paper presents five different semantic similarity methods. The methods proposed here understand the semantics associated with the word by making use of snippets returned by the Wikipedia or Britannica Encyclopedia for the given word pair. The snippets obtained are preprocessed. The preprocessing involves three different steps. First step is elimination of stop words. Second step is suffix removal & stemming . This task is achieved by applying Porter's Stemming Algorithm [2]. Third step involves keywords or index terms selection based on the frequency of occurrence of terms in the given snippet. In the proposed methods syntactic representation of the semantics associated word is achieved by following theses three steps. The set of keywords is used as syntactic representation of the snippet. Similarity between words is decided using this set of keywords.

II. PROPOSED SEMANTIC SIMILARITY METHOD

A. Snippet Extraction

Wikipedia is the world's largest collaboratively edited source of encyclopedic knowledge. It provides semantic information for every word or term. Semantics associated with each word is very well described by Wikipedia. Firstly, we must decide which part in Wikipedia for a word is useful for us. For example, if we search word "*car*" in Wikipedia, we can get much information about "*car*", such as car's history, its production and its safety, and so on. Use of this complete information may mislead the task of deciding semantic similarity. Usually, Wikipedia return some top result for the word for which we search information in Wikipedia. These snippets use simple vocabulary to explain the word, or give simple definition or some description about the word. These snippets are very much suitable to measure semantic similarity between words.

B. Snippet Preprocessing

The snippets downloaded from Wikipedia can not be directly used. There are lot of semantically unrelated words. Also the words in different form may bring in negative impact on similarity computation. So preprocessing of snippets is needed. Preprocessing of snippets involves three steps: removal of high frequency words, suffix stripping, detecting equivalent stems.

Stop Word Removal

Luhn [1] proposed that *"the frequency of word occurrence in an article furnishes a useful measurement of word significance*". Luhn used Zipf's Law [1] as a null hypothesis to specify two cut-offs, an upper and a lower (see Figure 2.), thus excluding non-significant words. The words exceeding the upper cut-off were considered to be common and those below the lower cut-off rare, and therefore not contributing significantly to the content of the article. He thus devised a counting technique for finding significant words.

The same is shown by using a plot of frequency versus rank:

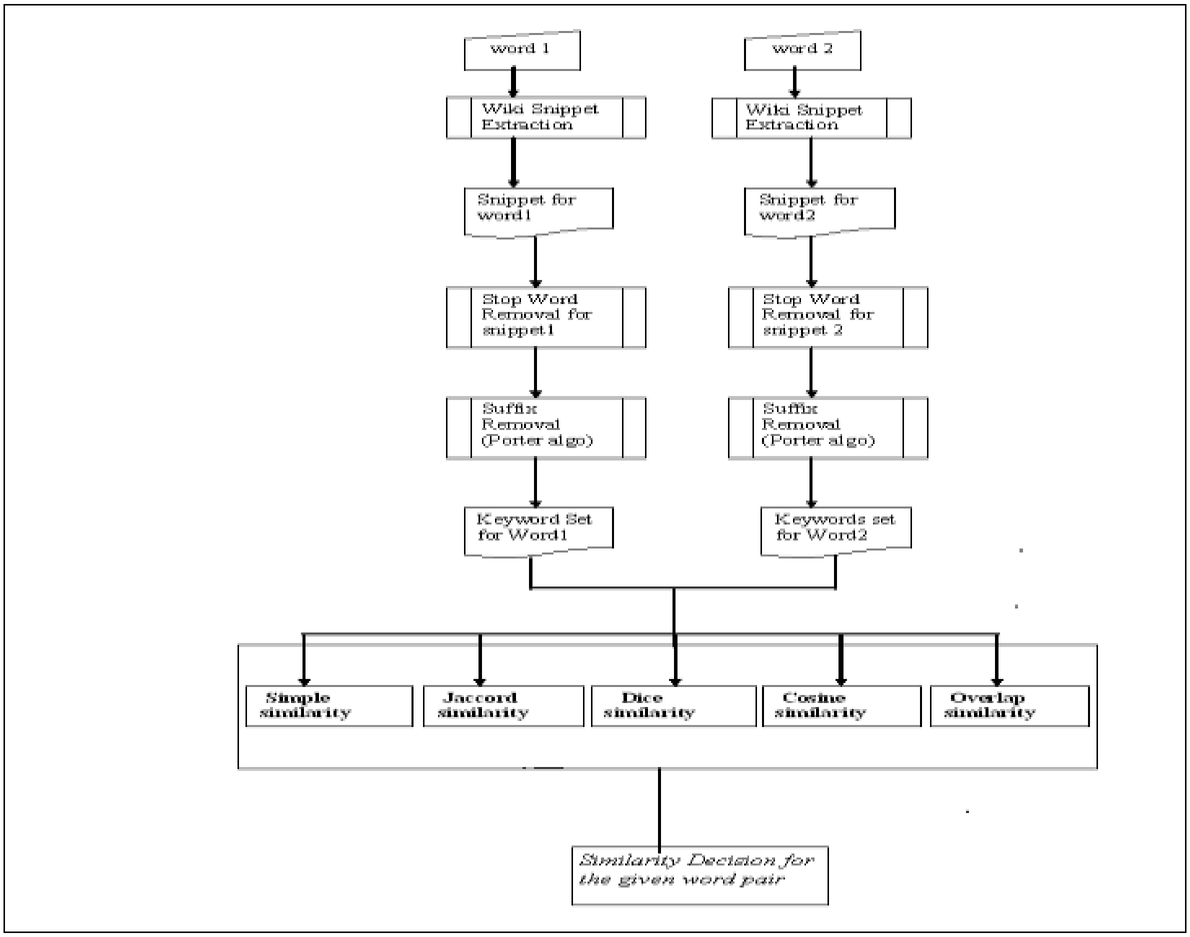


Figure 1: Flow of Similarity Computation Algorithm

The removal of high frequency words, 'stop' words or 'fluff' words is one way of implementing Luhn's upper cut-off. This is normally done by comparing the input text with a 'stop list' of words which are to be removed. The advantages of the process are non-significant words are removed so that they will not interfere during retrieval, also the size of the total text can be reduced by between 30 and 50 per cent.

Suffix Stripping And Stemming

Terms with a common stem will usually have similar meanings, for example: CONNECT, CONNECTED, CONNECTING, CONNECTION, CONNECTIONS. Performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, -IONS, etc to leave the single term CONNECT. In addition, the suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous.

Algorithm

An algorithm is proposed by M.F. Porter [2] for suffix stripping. Assumption for the algorithm is: a '*consonant*' in a word is: "*a letter other than A, E, I, O or U, and other than Y preceded by a consonant".*

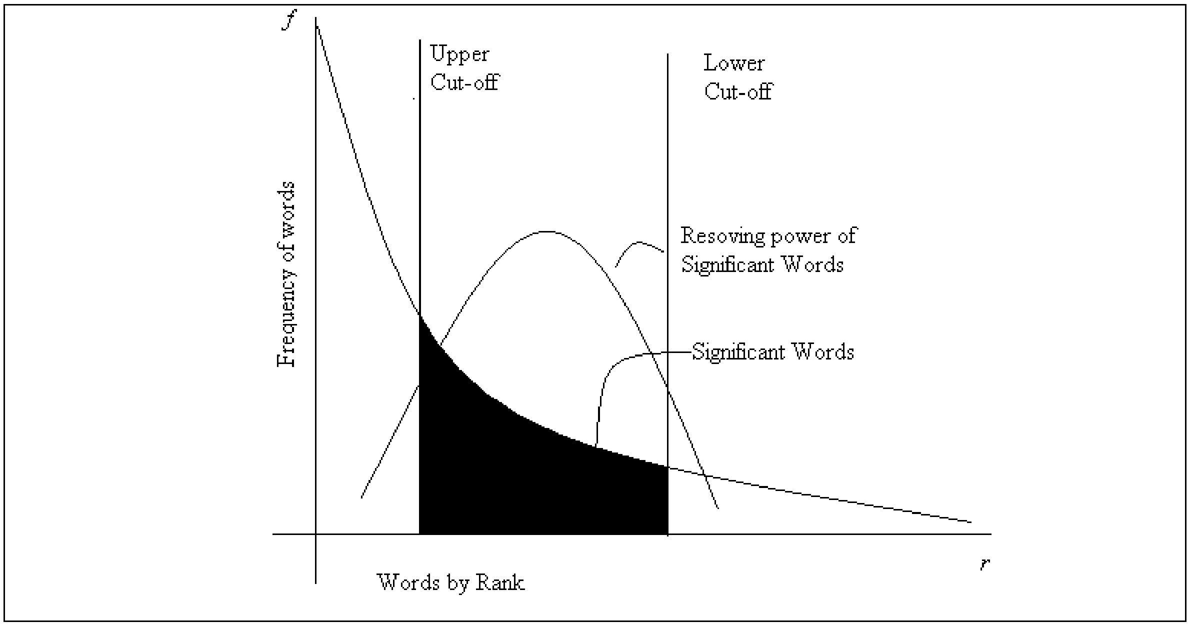


Figure 2 : Relation between frequency of word and significance of word [1]

A *'vowel'* in a word is :"*if a letter is not a consonant it is a vowel"*. Every consonant is represented by 'C' and every vowel is represented by 'V'. A list CCC.... of length greater than 0 will be denoted by C, and a list VVV... of length greater than 0 will be denoted by V. Any word, or part of a word, therefore has one of the four forms:

CVCV ... C
CVCV ... V
VCVC ... C
VCVC ... V

These all may be represented by the single form : [C]VCVC ... [V]. Where, the square brackets denote arbitrary presence of their contents. Using (VC){m} to denote VC repeated m times, this may again be written as :

[C](VC){m}[V]

'm' will be called the 'measure' of any word or word part when represented in this form. Here are some examples:

m=0 TREE, ME, BY.
m=1 TROUBLE, OATS, TREES, IVY.
m=2 TROUBLES, PRIVATE, OATEN, ORRERY.

The 'rules' for removing a suffix will be given in the form:

$$(\text{condition})\ S_1 \rightarrow S_2$$

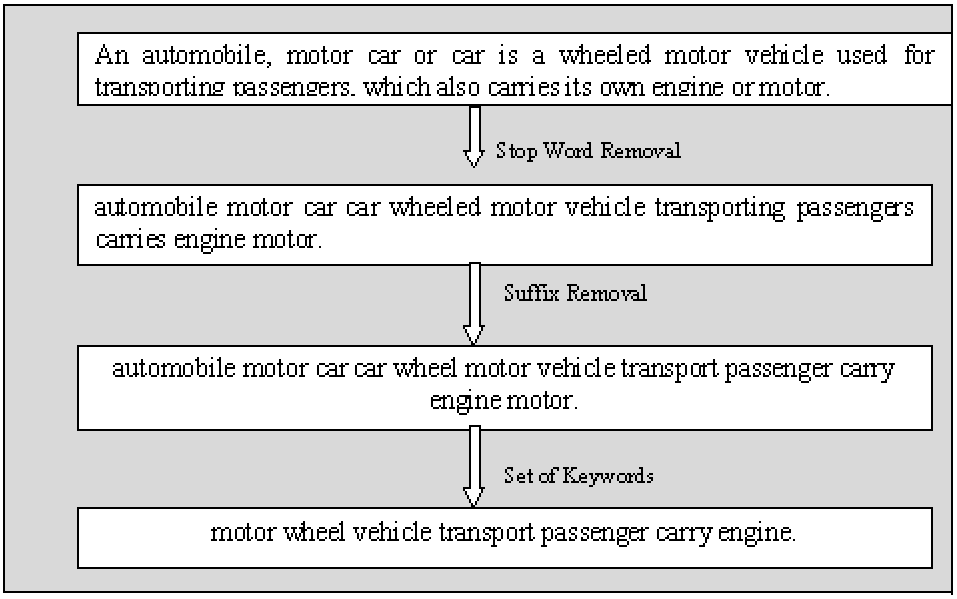
This means that if a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2. The condition is usually given in terms of m, e.g.:

$$(m > 1)\ \text{EMENT} \rightarrow$$

Here S1 is `EMENT' and S2 is null. This would map REPLACEMENT to REPLAC, since REPLAC is a word part for which m = 2.

TRANSPORTING
CCVCCCVCCVCC
CVCVCVC
[C](VC)(3)

After stop word removal and suffix stripping, on the basis of frequency count of each term of the snippet, a set of keywords for the snippet is extracted. Figure given below explains the procedure of keyword extraction from the given Wikipedia snippet.



An automobile, motor car or car is a wheeled motor vehicle used for transporting passengers. which also carries its own engine or motor.

automobile motor car car wheeled motor vehicle transporting passengers carries engine motor.

automobile motor car car wheel motor vehicle transport passenger carry engine motor.

motor wheel vehicle transport passenger carry engine.

Figure 3 : Extracting the Set of Keywords from the Wikipedia Snippet

C. Similarity Measures

Five different strategies are proposed in this paper to find out the semantic similarity results. Word pairs in Table 2 are used in investigating the suitability of individual strategies. For each of the proposed strategies, we carried out the experiments with two steps. Using the set of keywords, which are obtained from snippets by preprocessing them, semantic similarity values of the word pairs are calculated. Then, the correlation coefficient between the computed semantic similarity values and the human ratings of Rubenstein-Goodenough's is calculated. This correlation coefficient is used to judge the suitability of the particular strategy comparing to other strategies and previously published results.

The five similarity measures proposed here are based on the five commonly used measures of association in information retrieval. Snippets used here are represented by a set of keywords and the counting measure **| . |** gives the size of the set. For the word w_1, K| w_1 | is the set of keywords obtained from snippet and for the word w_2, K| w_2 | is the set of keywords obtained from the snippet.

Strategy 1: The Strategy 1 is based on *Jaccard index*, also known as the Jaccard similarity coefficient is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(w_1, w_2) = \frac{|K|w_1| \cap K|w_2||}{|K|w_1| + K|w_2| - K|w_1| \cap K|w_2||} \tag{6}$$

Strategy 2: The Strategy 2 is based on *Dice's coefficient*, named after Lee Raymond Dice and also known as the Dice coefficient. It is a similarity measure related to the Jaccard index. For sets *X* and *Y* of keywords used in information retrieval, the coefficient may be defined as:

$$D(w_1, w_2) = 2\frac{|K|w_1| \cap K|w_2||}{|K|w_1| + K|w_2||} \tag{7}$$

Strategy 3: Strategy 3 is based on Overlap Coefficient.The *overlap coefficient* is a similarity measure related to the Jaccard index that computes the overlap between two sets. If set *X* is a subset of *Y* or the converse then the overlap coefficient is equal to one.

$$O(w_1, w_2) = \frac{|K|w_1| \cap K|w_2||}{\min(|K|w_1||, |K|w_2||)} \tag{8}$$

Strategy 4: Strategy 4 is based on Cosine Similarity measure. *The Cosine similarity* is a measure of similarity between two vectors of *n* dimensions by finding the angle between them. It is often used to compare documents in text mining.

$$C(w_1, w_2) = \frac{|K|w_1| \cap K|w_2||}{sqrt(|K|w_1||) \times sqrt(|K|w_2||)} \tag{9}$$

Strategy 5 : Strategy 5 based on Simple matching coefficient, which is the number of shared index terms. This coefficient does not take into account the sizes of *X* and *Y*.

The following coefficients which have been used in document retrieval take into account the information provided by the sizes of *X* and *Y*. $S(w_1, w_2) = |K|w_1| \cap K|w_2||$

(10)

III. RESULTS

TABLE 1 : Comparison of Similarity Methods for Miller-Charles Data Set

Method	Type						Correlat
		Word Ontology	Corp	Search engine	Page Count	Wikipedia/ Encyclopedia	
Edge counting	Edge Counting	✓	×	×	×	×	0.664
Information Content	Information Content	×	✓	×	×	×	0.745
Jiang & Conarth	Hybrid	✓	✓	×	×	×	0.848
Lin	Information Content	×	✓	×	×	×	0.821
Yuhua Li	Hybrid	✓	✓	×	×	×	0.891
WebSim By Danushka	Web Based	×	×	×	✓	×	0.834
Google Similarity	Web Based	×	×	×	✓	×	0.66
Relational Sim By Danushka	Web Based	×	×	✓	×	×	0:834
Elias Iosif	Web Based	×	×	✓	×	×	0.88
Proposed Measures	Web Based	×	×	×	×	✓	0.80

For deciding whether a specific method has performed better or has not, we calculate the correlation coefficient of the semantic similarity results of the method and human judgment for the benchmark dataset. For two datasest X and Y correlation coefficient is computed by:

$$\rho = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathrm{X_i} - \mu_x}{\sigma_x}\right)\left(\frac{Y_{\mathrm{i}} - \mu_y}{\sigma_y}\right) \qquad (11)$$

Performance of semantic similarity methods proposed here is assessed by making use of benchmark datasets given by Rubenstein- Goodenough [9] and a word set given by Miller and Charles [9]. Rubenstein- Goodenough's Benchmark dataset consists of 65 word pairs. These 65 word pairs are divided into sets called as dataset D_0 and D_1. The dataset D_0 is utilized by Miller and Charles in his experiment. They have rated similarities between words from "0 to 4". "0" – semantically unrelated and "4" – highly similar / highly synonymous. Before presenting the achieved results of the above mentioned five strategies the Table 2 lists various similarity methods.

IV. CONCLUSION

This paper presents a new approach for measuring semantic similarity between words using the Snippets returned by Wikipedia and the five different similarity measures of association. Snippets in Wikipedia are used to measure semantic similarity between words. The result demonstrates that the snippets in Wikipedia have a significant influence on the accuracy of semantic similarity measure between words.

Table 1 summarizes various similarity methods and compares the approaches followed by them. Table 2 gives results of five different similarity methods proposed in this paper. Table 3 summarizes the correlation coefficient of the proposed methods using MC replica and RG ratings

The major contributions of this paper are:

1. Measuring semantic similarity between words using Keywords obtained from Wikipedia Snippets is proposed in this paper.
2. Luhn's idea for deciding the significant words is applied for preprocessing of snippets.

3. Porter's algorithm is used for suffix removal in preprocessing snippets.
4. Five association measures of Jaccord, Dice, Overlap cosine and simple matching are used to measure semantic similarity between words.

TABLE 2 : Similarity Results from Different Measures on Miller Charle's Benchmark Dataset

WORD PAIR	**RG** **Rating**	**MC** **Replica**	**Resnik** **Replica**	**S1**	**S2**	**S3**	**S4**	**S5**
cord-smile	0.02	0.13	0.1	0.00	0.00	0.00	0.00	0.00
Rooster-voyage	0.04	0.08	0	0.00	0.00	0.00	0.00	0.00
noon-string-	0.04	0.08	0	0.00	0.00	0.00	0.00	0.00
glass-magician	0.44	0.11	0.1	0.00	0.00	0.00	0.00	0.00
Monk-slave	0.57	0.55	0.7	0.00	0.00	0.00	0.00	0.00
coast-forest	0.85	0.42	0.6	0.44	0.80	1.33	0.87	1.00
monk-oracle	0.91	1.1	0.8	0.40	0.73	0.80	0.73	1.00
lad-wizard	0.99	0.42	0.7	0.80	1.33	1.33	1.33	1.00
forest- graveyard	1.00	0.84	0.6	0.31	0.57	0.57	0.57	1.00
food-rooster	1.09	0.89	1.1	0.00	0.00	0.00	0.00	0.00
coast-hill	1.26	0.87	0.7	0.44	0.80	1.33	0.87	1.00
car-journey	1.55	1.16	0.7	0.44	0.89	1.33	0.94	1.00
crane-implement	2.37	1.68	0.3	0.86	2.18	4.00	2.45	3.00
brother-lad	2.41	1.66	1.2	1.71	3.43	4.00	3.46	3.00
bird-crane	2.63	2.97	2.1	0.00	0.00	0.00	0.00	0.00
bird -cock	2.63	3.05	2.2	0.00	0.00	0.00	0.00	0.00
Food-fruit	2.69	3.08	2.1	0.00	0.00	0.00	0.00	0.00
Brother-monk	2.74	2.82	2.4	0.00	0.00	0.00	0.00	0.00
asylum-madhouse	3.04	3.61	3.6	4.00	4.00	4.00	4.00	2.00
furnace-stove	3.11	3.11	2.6	1.33	2.00	2.00	2.00	1.00
magician-wizard	3.21	3.5	3.5	4.00	4.00	4.00	4.00	1.00
Journey-voyage	3.58	3.84	3.5	4.00	4.00	4.00	4.00	3.00
coast-shore	3.60	3.7	3.5	4.00	4.00	4.00	4.00	2.00
implement-tool	3.66	2.95	3.4	4.00	4.00	4.00	4.00	3.00
Boy-lad	3.82	3.76	3.5	3.00	4.00	4.00	4.00	3.00
automobile-car	3.92	3.92	3.9	4.00	4.00	4.00	4.00	5.00
midday-noon	3.94	3.42	3.6	4.00	4.00	4.00	4.00	6.00
gem-jewel	3.94	3.84	3.5	4.00	4.00	4.00	4.00	4.00

TABLE 3 :Correlation of Different Strategies against Human Similarity Judgements on Benchmark Dataset

Strategy	**S_1**	**S_2**	**S_3**	**S_4**	**S_5**

Correlation (MC Replica)	0.8006	0.7958	0.7465	0.7910	0.6401
Correlation (RG Rating)	0.7974	0.7955	0.7609	0.7934	0.6968

REFERENCES

[1] C.J. Rijsbergen, "Information Retrieval", (www.dcs.gla.ac.uk)

[2] M.F.Porter, “An algorithm for suffix stripping”, Originally published in \Program\, \14\ no. 3, pp 130-137, July 1980.

[3] J. Jiang and D. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy,” in *Proc. of International Conference on Research on Computational Linguistics*, 1997.

[4] Yuhua Li, Zuhair A. Bandar and David McLean, “An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources”, IEEE Transactions On Knowledge And Data Engineering, Vol. 15,No.4,July/August2003

[5] J. Gracia, R. Trillo, M. Espinoza, and E. Mena, “Querying the web: A multiontology disambiguation method,” in *Proc. of International Conference on Web Engineering*, 2006, pp. 241–248

[6] D. Bollegala, Y. Matsuo, and M. Ishizuka, “Measuring semantic similarity between words using web search engines,” in *Proc. of International Conference on World Wide Web*, 2007, pp. 757–766.

[7] Rudi L. Cilibrasi and Paul M.B. Vita´ nyi, “The Google Similarity Distance”, IEEE Transactions On Knowledge And Data Engineering, Vol. 19, No. 3, March 2007

[8] Elias Iosif and Alexandros Potamianos, “Unsupervised Semantic Similarity Computation Between Terms Using Web Documents”, IEEE Transactions On Knowledge And Data Engineering

[9] Lu Zhiqiang, Shao Werimin and Yu Zhenhua, “Measuring Semantic Similarity between Words Using Wikipedia”, 2009 International Conference on Web Information Systems and Mining