Measuring Semantic Similarity between Words Using Web Pages

T.Sujatha, Ramesh Naidu G, P.Suresh B

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 cannot 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: *Pre-compiled database based metrics*, i.e., metrics consulting only human-built knowledge resources, such as onto logies, *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 *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.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 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 proposed a page-count-based similarity measure, called the Normalized Google Distance.

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

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

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

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 (*ti*) in document *dj*, and the total number of occurrences of all terms in document *dj*.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad \text{------}$$

3

Manuscript received on July, 2012.

T.Sujatha, Department of CSE, Kaushik College of Engineering, Vishakhapattanam (A.P.), India
Prof. Ramesh Naidu G, Department of CSE, Kaushik College of Engg., Vishakhapattanam (A.P.), India
Prof. P.Suresh Babu, Department of CSE, Kaushik College of Engineering, Vishakhapattanam (A.P.), India

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.

$[t_{K,L}\ldots..t_{2,L}t_{1,L}]w[t_{1,R}\ldots..t_{2,R}t_{K,R}]$ where,

$t_{i,L}$ and $t_{i,R}$ represent the ith word to the left and to the right of w respectively.

Each word is represented as a feature vector as $F_{w,k}=(v_{w,1}, v_{w,2},\ldots. v_{w,N})$.There are various feature weighting schemes for computing the value of $V_{w,i}$, 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

II. PROPOSED SEMANTIC SIMILARITY METHOD

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 simðP;QÞ that returns a value in range [0,1] If P and Q are highly similar (e.g.,synonyms), we expect simðP;QÞ to be closer to 1. On the other hand if P and Q are similar, then we expect simðP;QÞ 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. Using this feature representation of words, we train a two-class support vector machine to classify synonymous and nonsynonymous word pairs. The function simðP;QÞ is then approximated by the confidence score of the trained SVM.

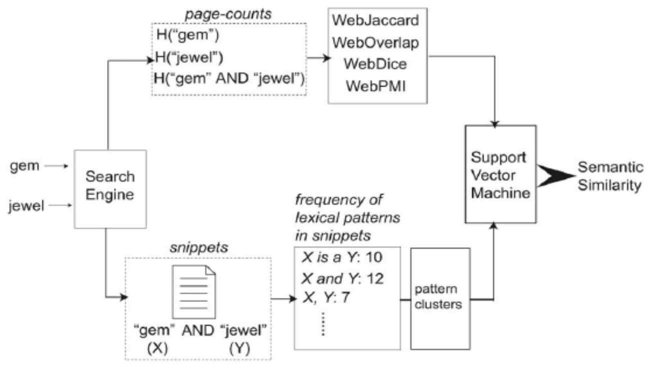


Fig.1. Outline of the proposed method

Fig. 1 illustrates an example of using the proposed method to compute the semantic similarity between two words, gem and jewel. First, we query a web search engine and retrieve page counts for the two words and for their conjunctive (i.e., "gem," "jewel," and "gem AND jewel"). In Section 2.2, we define four similarity scores using page counts. Page counts-based similarity scores consider the global co-occurrences of two words on the web. However, they do not consider the local context in which two words co-occur. On the other hand, snippets returned by a search engine represent the local context in which two words co-occur on the web. Consequently, we find the frequency of numerous lexical syntactic patterns in snippets returned for the conjunctive query of the two words. The lexical patterns we utilize are extracted automatically using the method described in Section 2.3. However, it is noteworthy that a semantic relation can be expressed using more than one lexical pattern. Grouping the different lexical patterns that convey the same semantic relation, enables us to represent a semantic relation between two words accurately. For this purpose, we propose a sequential pattern clustering algorithm in Section 2.4. Both page counts-based similarity scores and lexical pattern clusters are used to define various features that represent the relation between two words. Using this feature representation of word pairs, we train a two-class support vector machine in Section 2.5.

2.2 Page Count-Based Co-Occurrence Measures

Page counts for the query P AND Q can be considered as an approximation of co-occurrence of two words (or multiword phrases) P and Q on the web. However, page counts for the query P AND Q alone do not accurately express semantic similarity. For example, Google returns 11,300,000 as the page count for "car" AND "automobile," whereas the same is 49,000,000 for "car" AND "apple." Although automobile is more semantically similar to car than apple is, page counts for the query "car" AND "apple" are more than four times greater than those for the query "car" AND "automobile." One must consider the page counts not just for the query P AND Q, but also for the individual words P and Q to assess semantic similarity between P and Q.

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 multiword phrases) P and Q, WebJaccard (P: Q), is defined as

WebJaccard (P: Q)

$$= \begin{cases} 0 & if\ H(P \cap Q) \le C \\ \dfrac{H(P \cap Q)}{H(P) + H(Q) - H(P \cap Q)} & other\ wis \end{cases}$$

Therein, P ∩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,weset the WebJaccard coefficient to zero if the page count for the queryP∩ Qis less than a threshold c.2 Similarly, we define WebOverlap, WebOverlap (P,Q) as

WebOverlap (P,Q)

$$= \begin{cases} 0 & if\ H(P \cap Q) \le C \\ \dfrac{H(P \cap Q)}{min(H(P), H(Q))} & other\ wis \end{cases}$$

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

WebDice(P:Q)

$$= \begin{cases} 0 & if\ H(P \cap Q) \le C \\ \dfrac{2H(P \cap Q)}{H(P) + H(Q)} & other\ wis \end{cases}$$

Pointwise mutual information [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 point wise mutual information using page counts as WebPMI(P:Q)

$$= \begin{cases} 0 & \text{if } H(P \cap Q) \leq C \\ \log_2\left(\dfrac{\frac{H(P \cap Q)}{N}}{\frac{H(P)}{N}\,\frac{H(Q)}{N}}\right) & \text{other wise} \end{cases}$$

Here, N is the number of documents indexed by the search engine. Probabilities in (4) are estimated according to the maximum likelihood principle. To calculate PMI accurately using (4), we must know N, the number of documents indexed by the search engine. Although estimating the number of documents indexed by a search engine is an interesting task itself, it is beyond the scope of this work. In the present work,

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

Fig 2.. A snippet retrieved for the query "cricket" AND "sport."

we set N =1010according to the number of indexed pages reported by Google. As previously discussed, page counts are mere approximations to actual word co-occurrences in the web. However, it has been shown empirically that there exists a high correlation between word counts obtained from a web search engine (e.g., Google and Altavista) and that from a corpus (e.g., British National corpus) Moreover, the approximated page counts have been successfully used to improve a variety of language modeling tasks .

2.3 Lexical Pattern Extraction

Page counts-based co-occurrence measures described in Section 2.2 do not consider the local context in which those words co-occur. This can be problematic if one or both words are polysemous, or when page counts are unreliable.

On the other hand, the snippets returned by a search engine for the conjunctive query of two words provide useful clues related to the semantic relations that exist between two words. 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. For example, consider the snippet in Fig. 2. Here, the phrase is a indicates a semantic relationship between cricket and sport.

Many such phrases indicate semantic relationships. For example, also known as, is a, part of, is an example of all indicate semantic relations of different types. 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. Despite the efficiency of using snippets, they pose two main challenges: first, a snippet can be a fragmented sentence, second, a search engine might produce a snippet by selecting multiple text fragments from different portions in a document. Because most syntactic or dependency parsers assume complete sentences as the input, deep parsing of snippets produces incorrect results. Consequently, we propose a shallow lexical pattern extraction algorithm using web snippets, to recognize the semantic relations that exist between two words. Lexical syntactic patterns have been used in various natural language processing tasks such as extracting hypernyms or meronyms, question answering , and paraphrase extraction .

Ostrich,a large,flightless bird that lives in the dry grasslands of africa

Fig. 3. A snippet retrieved for the query "ostrich ….bird."

Although a search engine might produce a snippet by selecting multiple text fragments from different portions in a document, a predefined delimiter is used to separate the different fragments. For example, in Google, the delimiter "..." is used to separate different fragments in a snippet.

We use such delimiters to split a snippet before we run the proposed lexical pattern extraction algorithm on each fragment.

Given two words P and Q, we query a web search engine using the wildcard query "P …Q" and download snippets. The "." operator matches one word or none in a webpage. Therefore, our wildcard query retrieves snippets in which P and Q appear within a window of seven words.

Because a search engine snippet contains ca. 20 words on average, and includes two fragments of texts selected from a document, we assume that the seven word window is sufficient to cover most relations between two words in snippets. In fact, over 95 percent of the lexical patterns extracted by the proposed method contain less than five words. We attempt to approximate the local context of two words using wildcard queries. For example, Fig. 3 shows a snippet retrieved for the query "ostrich…. bird."

For a snippet δ, 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 δ that satisfy all of the following conditions:

1. A subsequence must contain exactly one occurrence of each X and Y.

2. The maximum length of a subsequence is L words.

3. A subsequence is allowed to skip one or more

Words. However, we do not skip more than g

Number of words consecutively. Moreover, the total number of words skipped in a subsequence should not exceed G.

4. We expand all negation contractions in a context. For example, didn't is expanded to did not. 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. Finally, we count the frequency of all generated subsequences and only use subsequences that occur more than T times as lexical patterns.

The parameters L,g,G, and T are set experimentally, It is noteworthy that the Proposed pattern extraction algorithm considers all the words in a snippet, and is not limited to extracting patterns only from the mid fix (i.e., the portion of text in a snippet that appears between the queried words). Moreover, the Consideration of gaps enables us to capture relations between distant words in a snippet. We use a modified version of the prefix span algorithm to generate subsequences From a text snippet. Specifically, we use the Constraints (2-4) to prune the search space of candidate subsequences. For example, if a subsequence has reached the maximum length L, or the number of skipped words is G, then we will not extend it further. By pruning the search space, we can speed up the pattern generation process.

However, none of these modifications affect the accuracy of the proposed semantic similarity measure because the modified version of the prefixspan algorithm still generates the exact set of patterns that we would obtain if we used the original prefixspan algorithm (i.e., without pruning) and subsequently remove patterns that violate the above mentioned constraints. For example, some patterns extracted from the snippet shown in Fig. 3 are: X, a large Y, X a flightless Y, and X, large Y lives.

2.3 Snippets

It is a brief window of text extracted by a search engine around the query term in a document.It provides useful information regarding the local context of the query term. 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, personal name disambiguation], and community mining . Processing snippets is also efficient because it obviates the trouble of downloading webpages, which might be time consuming depending on the size of the pages. However, a widely acknowledged drawback of using snippets is that, because of the huge scale of the web and the large number of documents in the result set, only those snippets for the topranking results for a query can be processed efficiently. Ranking of search results, hence snippets, is determined by a complex combination of various factors unique to the underlying search engine. Therefore, no guarantee exists that all the information we need to measure semantic similarity between a given pair of words is contained in the top-ranking snippets. .

Drawback Because of the huge scale of the web and the large no. of documents in the results set, only those snippets for the top ranking results for a query can be processed efficiently.

2.5 Lexical Pattern 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. According to the distributional hypothesis, words that occur in the same context have similar meanings. The distributional hypothesis has been used in various related tasks, such as identifying related words , and extracting paraphrases . If we consider the word pairs that satisfy (i.e., co-occur with) a particular lexical pattern as the context of that lexical pair, then from the distributional hypothesis, it follows that the lexical patterns which are similarly distributed over word pairs must be semantically similar.

We represent a pattern a by a vector a of word-pair frequencies. We designate a, the word-pair frequency vector of pattern a. It is analogous to the document frequency vector of a word, as used in information retrieval. The value of the element corresponding to a word pair (P:Q)in a, is the frequency, f(PiQi,a), that the pattern a occurs with the word pair(Pi:Qi). As demonstrated later, the proposed pattern extraction algorithm typically extracts a large number of lexical patterns. Clustering algorithms based on pairwise comparisons among all patterns are prohibitively time consuming when the patterns are numerous. Next, we present a sequential clustering algorithm to efficiently cluster the extracted patterns.

Given a set A of patterns and a clustering similarity Threshold θ, Algorithm 1 returns clusters (of patterns) that express similar semantic relations. First, in Algorithm 1, the function SORT sorts the patterns into descending order of their total occurrences in all word pairs. The total occurrence μ(a) of a pattern a is the sum of frequencies over all word pairs, and is given by μ(a)=f(Pi,Qi,a).

After sorting, the most common patterns appear at the beginning in Λ, whereas rare patterns (i.e., patterns that occur with only few word pairs) get shifted to the end. Next, in line 2, we initialize the set of clusters, C, to the empty set. The outer for loop (starting at line 3), repeatedly takes a pattern ai from the ordered set Λ, and in the inner for loop (starting at line 6), finds the cluster, C*(Є C) that is most similar to ai. First, we represent a cluster by the centroid of all word-pair frequency vectors corresponding to the patterns in that cluster to compute the similarity between a pattern and a cluster. Next, we compute the cosine similarity between the cluster centroid (Cj), and the word-pair frequency vector of the pattern (aj). If the similarity between a pattern ai, and its most similar cluster, C*, is greater than the threshold θ, we append ai to C*(line14). We use the operator $\oplus$ to denote the vector addition between C* and ai. Then, we form a new cluster {ai} and append it to the set of clusters, C, if ai is not similar to any of the existing clusters beyond the threshold θ.

By sorting the lexical patterns in the descending order of their frequency and clustering the most frequent patterns first, we 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. The greedy sequential nature of the algorithm avoids pairwise comparisons between all lexical patterns. This is particularly important because when the number of lexical patterns is large as in our experiments (e.g., over 100,000), pairwise comparisons between all patterns are computationally prohibitive. The proposed clustering algorithm attempts to identify the lexical patterns that are similar to each other more than a given threshold value. By adjusting the threshold, we can obtain clusters with different granularity.

Algorithm 1: Sequential pattern clustering algorithm

```
Input: patterns $\Lambda$={a1,….an},threshold θ
Output: clusters C
1: SORT (Λ)
2: C $\leftarrow$ {}
```

```
3: for pattern ai $\in$ Λ do
4: max $\leftarrow$ -$\infty$
5:     c* $\leftarrow$ null
6: for cluster cj $\in$ $C$ do
7:     sim $\leftarrow$ cosine(ai,cj)
8:     if sim>max then
9:     max $\leftarrow$ sim
10:    c* $\leftarrow$ cj
11:    end if
12:    end for
13:    if max > θ then
14:    c*$\leftarrow$ c* $\oplus$ {ai}
15:    else
16:    C←C $\cup$ {ai}
17:  end if
18:  end for
19:  return C
```

The only parameter in Algorithm 1, the similarity threshold, θ ranges in [0,1]. It decides the purity of the formed clusters. Setting θ to a high value ensures that the patterns in each cluster are highly similar. However, high θ values also yield numerous clusters (increased model complexity). the effect of θ on the overall performance of the proposed relational similarity measure.

The initial sort operation in Algorithm 1 can be carried out in time complexity of O(nlogn) where n is the number of patterns to be clustered. Next, the sequential assignment of lexical patterns to the clusters requires complexity ofO(n|c|), where |C| is the number of clusters. Typically, n is much larger than |c|. Therefore, the overall time complexity of Algorithm 1 is dominated by the sort operation, hence O(nlogn). The sequential nature of the algorithm avoids pairwise comparisons among all patterns.

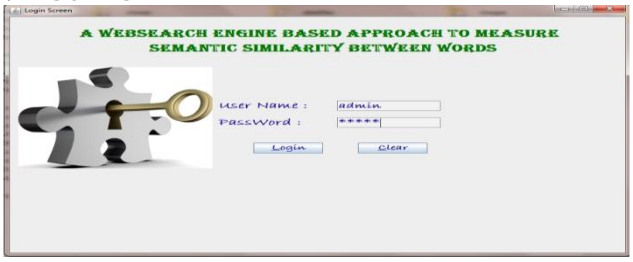
Moreover, sorting the patterns by their total word-pair frequency prior to clustering ensures that the final set of clusters contains the most common relations in the data set.

2.6 SVM (Support Vector Machine)

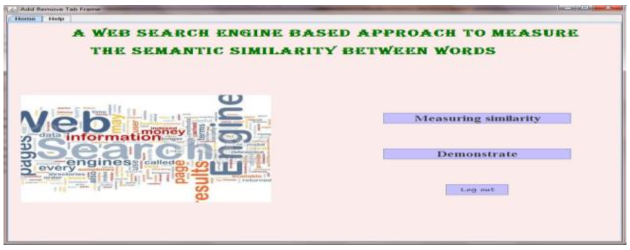
SVMs are currently among the best performers for a number of classification tasks ranging from text to genomic data SVMs can be applied to complex data types beyond feature vectors (e.g. graphs, sequences, and relational data) by designing kernel functions for such data.SVM was trained using page count co-occurrence measures, lexical pattern clustering & snippets to extract the synonymous & non-synonymous word pairs which give semantic similarity.

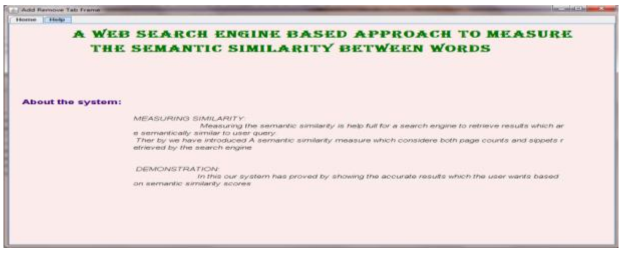
To train the two-class SVM. We require both synonymous and nonsynonymous word pairs. We use WordNet, a manually created English dictionary, to generate the training data required by the proposed method. For each sense of a word, a set of synonymous words is listed in WordNet synsets. We randomly select 3,000 nouns from WordNet, and extract a pair of synonymous words from a synset of each selected noun. If a selected noun is polysemous, then we consider the synset for the dominant sense. Obtaining a set of nonsynonymous word pairs (negative training instances) is difficult, because there does not exist a large collection of manually created nonsynonymous word pairs. Consequently, to create a set of nonsynonymous word pairs, we adopt a random shuffling technique. Specifically, we first rand omly select two synonymous word pairs from the set of synonymous word pairs created above, and exchange two words between word pairs to create two new word pairs. For example, from two synonymous word pairs A;B and C;D, we generate two new pairs A;C and B;D. If the newly created word pairs do not appear in any of the word net synsets, we select them as nonsynonymous word pairs. We repeat this process until we create 3,000 nonsynonymous word pairs. Our final training data set contains 6,000 word pairs (i.e., 3,000 synonymous word pairs and 3,000 nonsynonymous word pairs). Next, we use the lexical pattern extraction algorithm to extract numerous lexical patterns for the word pairs in our training data set. We experimentally set the parameters in the pattern extraction algorithm to L ¼ 5, g ¼ 2, G ¼ 4, and T ¼ 5. the number of patterns extracted for synonymous and nonsynonymous word pairs in the training data set. As can be seen from Table 1, the proposed pattern extraction algorithm typically extracts a large number of lexical patterns.Because of the noise in web snippets such as, ill-formed snippets and misspells, most patterns occur only a few times in the list of extracted patterns. Consequently, we ignore any patterns that occur less than five times. Finally, we deduplicate the patterns that appear for both synonymous and nonsynonymous word pairs to create a final set of 3,02,286 lexical patterns. The remainder of the experiments described in the paper use this set of lexical patterns

3.RESULTS


Login Screen
A WEBSEARCH ENGINE BASED APPROACH TO MEASURE SEMANTIC SIMILARITY BETWEEN WORDS
Login
Clear

User Name : admin
PassWord : *****

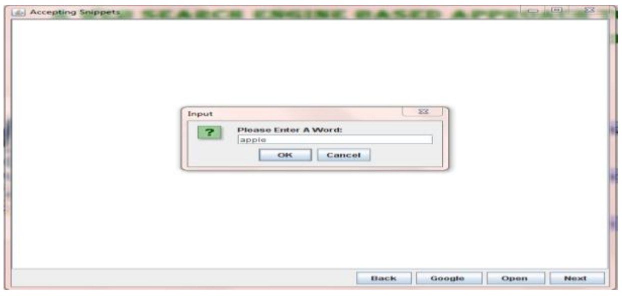

Add Remove Tab Frame
Home
Help
A WEB SEARCH ENGINE BASED APPROACH TO MEASURE THE SEMANTIC SIMILARITY BETWEEN WORDS
Measuring similarity
Demonstrate
Log out

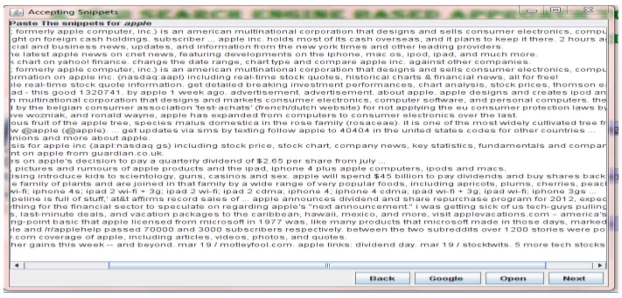

Add Remove Tab Frame
Home
Help
A WEB SEARCH ENGINE BASED APPROACH TO MEASURE THE SEMANTIC SIMILARITY BETWEEN WORDS

About the system:

MEASURING SIMILARITY:
Measuring the semantic similarity is help full for a search engine to retrieve results which ar e semantically similar to user query.
Ther by we have introduced A semantic similarity measure which considere both page counts and sippets r etrieved by the search engine

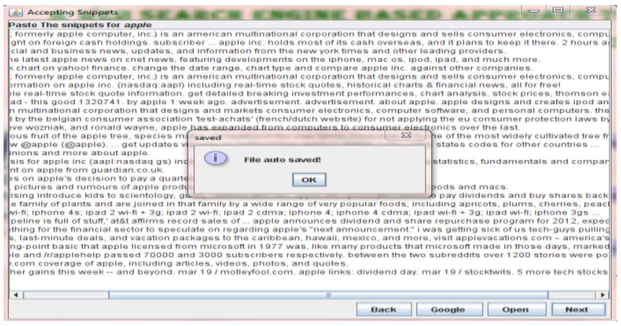
DEMONSTRATION:
In this our system has proved by showing the accurate results which the user wants based on semantic similarity scores





Paste The snippets for *apple*

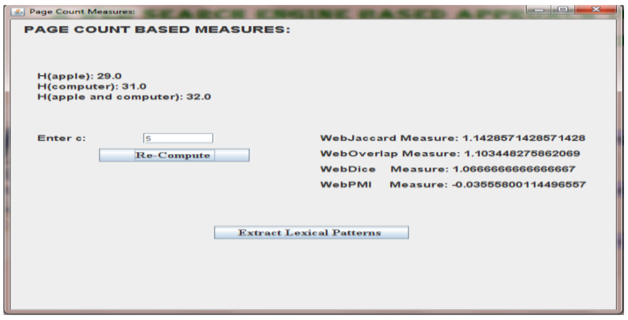
; formerly apple computer, inc.) is an american multinational corporation that designs and sells consumer electronics, compu
ght on foreign cash holdings. subscriber ... apple inc. holds most of its cash overseas, and it plans to keep it there. 2 hours a
cial and business news, updates, and information from the new york times and other leading providers.
ne latest apple news on cnet news, featuring developments on the iphone, mac os, ipod, ipad, and much more.
k chart on yahoo! finance. change the date range, chart type and compare apple inc. against other companies.
; formerly apple computer, inc.) is an american multinational corporation that designs and sells consumer electronics, compu
ormation on apple inc. (nasdaq:aapl) including real-time stock quotes, historical charts & financial news, all for free!
ole real-time stock quote information. get detailed breaking investment performances, chart analysis, stock prices, thomson e
ad - this good 1320741. by apple 1 week ago. advertisement. advertisement. about apple. apple designs and creates ipod an
n multinational corporation that designs and markets consumer electronics, computer software, and personal computers. the
l by the belgian consumer association 'test-achats' (french/dutch website) for not applying the eu consumer protection laws by
ve wozniak, and ronald wayne, apple has expanded from computers to consumer electronics over the last.
ous fruit of the apple tree, species malus domestica in the rose family (rosaceae). it is one of the most widely cultivated tree fr
w @apple (@apple). ... get updates via sms by texting follow apple to 40404 in the united states codes for other countries ...
inions and more about apple.
sis for apple inc (aapl:nasdaq gs) including stock price, stock chart, company news, key statistics, fundamentals and compar
nt on apple from guardian.co.uk.
es on apple's decision to pay a quarterly dividend of $2.65 per share from july ...
, pictures and rumours of apple products and the ipad, iphone 4 plus apple computers, ipods and macs.
ising introduce kids to scientology, guns, casinos and sex. apple will spend $45 billion to pay dividends and buy shares back
e family of plants and are joined in that family by a wide range of very popular foods, including apricots, plums, cherries, peac
wi-fi; iphone 4s; ipad 2 wi-fi + 3g; ipad 2 wi-fi; ipad 2 cdma; iphone 4; iphone 4 cdma; ipad wi-fi + 3g; ipad wi-fi; iphone 3gs ...
peline is full of stuff,' at&t affirms record sales of ... apple announces dividend and share repurchase program for 2012, expec
thing for the financial sector to speculate on regarding apple's "next announcement." i was getting sick of us tech-guys pulling
s, last-minute deals, and vacation packages to the caribbean, hawaii, mexico, and more, visit applevacations.com – america's
ng-point basic that apple licensed from microsoft in 1977 was, like many products that microsoft made in those days, marked
le and /r/applehelp passed 70000 and 3000 subscribers respectively. between the two subreddits over 1200 stories were po
.com coverage of apple, including articles, videos, photos, and quotes.
her gains this week -- and beyond. mar 19 / motleyfool.com. apple links: dividend day. mar 19 / stocktwits. 5 more tech stocks



Paste The snippets for *apple*

; formerly apple computer, inc.) is an american multinational corporation that designs and sells consumer electronics, compu
ght on foreign cash holdings. subscriber ... apple inc. holds most of its cash overseas, and it plans to keep it there. 2 hours a
cial and business news, updates, and information from the new york times and other leading providers.
ne latest apple news on cnet news, featuring developments on the iphone, mac os, ipod, ipad, and much more.
k chart on yahoo! finance. change the date range, chart type and compare apple inc. against other companies.
; formerly apple computer, inc.) is an american multinational corporation that designs and sells consumer electronics, compu
ormation on apple inc. (nasdaq:aapl) including real-time stock quotes, historical charts & financial news, all for free!
ole real-time stock quote information. get detailed breaking investment performances, chart analysis, stock prices, thomson e
ad - this good 1320741. by apple 1 week ago. advertisement. advertisement. about apple. apple designs and creates ipod an
n multinational corporation that designs and markets consumer electronics, computer software, and personal computers. the
l by the belgian consumer association 'test-achats' (french/dutch website) for not applying the eu consumer protection laws by
ve wozniak, and ronald wayne, apple has expanded from computers to consumer electronics over the last.
ous fruit of the apple tree, species m … e of the most widely cultivated tree fr
w @apple (@apple). ... get updates vi … states codes for other countries ...
inions and more about apple.
sis for apple inc (aapl:nasdaq gs) inc … statistics, fundamentals and compar
nt on apple from guardian.co.uk.
es on apple's decision to pay a quarte
, pictures and rumours of apple produ … pods and macs.
ising introduce kids to scientology, g … pay dividends and buy shares back
e family of plants and are joined in that family by a wide range of very popular foods, including apricots, plums, cherries, peac
wi-fi; iphone 4s; ipad 2 wi-fi + 3g; ipad 2 wi-fi; ipad 2 cdma; iphone 4; iphone 4 cdma; ipad wi-fi + 3g; ipad wi-fi; iphone 3gs ...
peline is full of stuff,' at&t affirms record sales of ... apple announces dividend and share repurchase program for 2012, expec
thing for the financial sector to speculate on regarding apple's "next announcement." i was getting sick of us tech-guys pulling
s, last-minute deals, and vacation packages to the caribbean, hawaii, mexico, and more, visit applevacations.com – america's
ng-point basic that apple licensed from microsoft in 1977 was, like many products that microsoft made in those days, marked
le and /r/applehelp passed 70000 and 3000 subscribers respectively. between the two subreddits over 1200 stories were po
.com coverage of apple, including articles, videos, photos, and quotes.
her gains this week -- and beyond. mar 19 / motleyfool.com. apple links: dividend day. mar 19 / stocktwits. 5 more tech stocks

File auto saved!

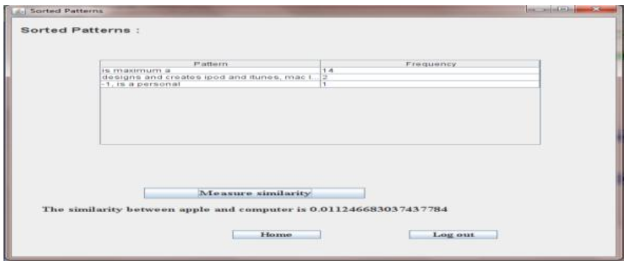


PAGE COUNT BASED MEASURES:

H(apple): 29.0
H(computer): 31.0
H(apple and computer): 32.0

Enter c: 5

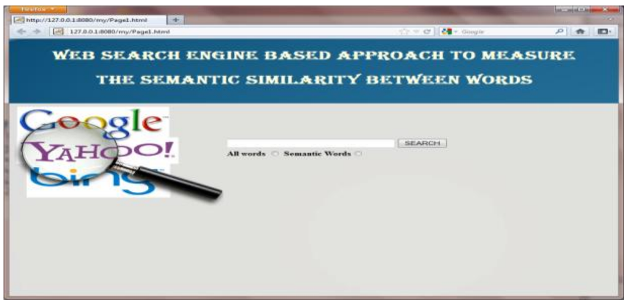
WebJaccard Measure: 1.1428571428571428
WebOverlap Measure: 1.103448275862069
WebDice Measure: 1.0666666666666667
WebPMI Measure: -0.0355580011449655 7

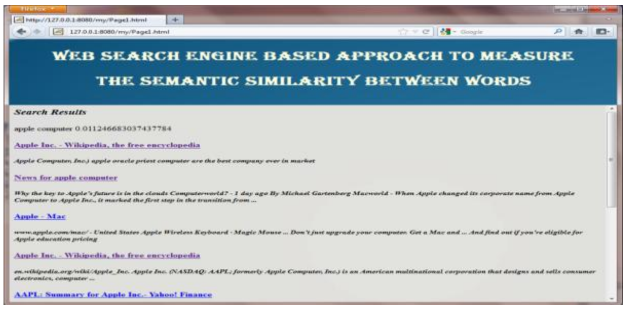


Sorted Patterns :

Pattern	Frequency
is maximum a	14
designs and creates ipod and itunes, mac i...	2
-1, is a personal	1

The similarity between apple and computer is 0.011246683037437784


http://127.0.0.1:8080/my/Page1.html
127.0.0.1:8080/my/Page1.html
WEB SEARCH ENGINE BASED APPROACH TO MEASURE
THE SEMANTIC SIMILARITY BETWEEN WORDS
Google
YAHOO!
bing
SEARCH
All words
Semantic Words


http://127.0.0.1:8080/my/Page1.html
127.0.0.1:8080/my/Page1.html
Google
WEB SEARCH ENGINE BASED APPROACH TO MEASURE
THE SEMANTIC SIMILARITY BETWEEN WORDS

Search Results

apple computer 0.011246683037437784

Apple Inc. - Wikipedia, the free encyclopedia

Apple Computer, Inc.) apple oracle priest computer are the best company ever in market

News for apple computer

Why the key to Apple's future is in the clouds Computerworld? - 1 day ago By Michael Gartenberg Macworld - When Apple changed its corporate name from Apple Computer to Apple Inc., it marked the first step in the transition from ...

Apple - Mac

www.apple.com/mac/ - United States Apple Wireless Keyboard - Magic Mouse ... Don't just upgrade your computer. Get a Mac and ... And find out if you're eligible for Apple education pricing

Apple Inc. - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/Apple_Inc. Apple Inc. (NASDAQ: AAPL; formerly Apple Computer, Inc.) is an American multinational corporation that designs and sells consumer electronics, computer ...

AAPL: Summary for Apple Inc.- Yahoo! Finance

Testing:

Overview:

The process of executing a system with the intent of finding errors

Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.

Quality is defined as justification of the requirements

Defect is nothing but deviation from the requirements.

Defect is nothing but bug.

Testing----the presence of bugs

Testing can demonstrate the presence of bugs, but not their absence

Debugging and Testing are not the same thing!

Testing is systematic attempt to break a program or the AUT

Debugging is the art or method of uncovering why the script/program did not execute properly.

Testing Methodologies:

Black box Testing: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.

Usually Test Engineers are involved in the black box testing.

White box Testing: is the testing process in which tester can perform testing on an application with having internal structural knowledge.

Usually the Developers are involved in white box testing.

Gray box Testing: is the process in which the combination of black box and white box tonics' are used.

4. Conclusion

We proposed a semantic similarity measure 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 two-class SVM was trained using those features extracted for synonymous and nonsynonymous word pairs selected from WordNet synsets. xperimental results on three benchmark data sets showed that the proposed method outperforms various baselines as well as previously proposed web-based semantic similarity measures, achieving a high correlation with human ratings.

REFERENCES

1. A. Kilgarriff, “Googleology Is Bad Science,” Computational Linguistics, vol. 33, pp. 147-151, 2007.
2. 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.
3. 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.
4. 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.
5. M. Hearst, “Automatic Acquisition of Hyponyms from Large Text Corpora,” Proc. 14th Conf. Computational Linguistics (COLING), pp. 539-545, 1992.
6. E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, “A Study on Similarity and Relatedness Using Distributional and Wordnet-Based Approaches,” Proc. Human Language Technologies: The 2009 Ann. Conf. North Am. Chapter of the Assoc. for Computational Linguistics (NAACL-HLT ’09), 2009.
7. G. Hirst and D. St-Onge, “Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms,” WordNet: An Electronic Lexical Database, pp. 305-332, MIT Press, 1998.
8. T. Hughes and D. Ramage, “Lexical Semantic Relatedness with Random Graph Walks,” Proc. Joint Conf. Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL ’07), pp. 581-589, 2007.
9. E. Gabrilovich and S. Markovitch, “Computing Semantic Relatedness Using Wikipedia-Based Explicit Semantic Analysis,” Proc. Int’l Joint Conf. Artificial Intelligence (IJCAI ’07), pp. 1606-1611, 2007.
10. Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka, “Polyphonet: An Advanced Social Network Extraction System,” Proc. 15th Int’l World Wide Web Conf., 2006.