Modified Pattern Extraction Algorithm for Efficient Semantic Similarity Measures between Words

Pushpa C N[1], Thriveni J[1], Venugopal K R[1] and L M Patnaik[2]

**[1] Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering
Bangalore, Karnataka, India**

**[2] Indian Institute of Science,
Bangalore, Karnataka, India**

Abstract

Semantic Similarity measures plays an important and significant role in information retrieval, natural language processing and various tasks on web such as relation extraction, community mining, document clustering, and automatic meta-data extraction. In this paper, we have proposed a Modified Pattern Extraction Algorithm [MPEA] to compute the semantic similarity measure between the words by combining both page count method and web snippets method. Four association measures are used to find semantic similarity between words in page count method using web search engines. We use a Sequential Minimal Optimization (SMO) Support Vector Machines (SVM) to find the optimal combination of page counts-based similarity scores and top-ranking patterns from the web snippets method. The SVM is trained to classify the synonymous word-pairs and non-synonymous word-pairs. The proposed approach aims to improve the Correlation values, Precision, Recall, and F-measures, compared to the existing methods. The proposed algorithm outperforms by 89.8 % of correlation value for Miller-Charles dataset and 75.3% of correlation value for Word Similarity dataset.

Keywords: *Information Retrieval, Semantic Similarity, Support Vector Machine, Web Mining, Web Search Engine, Web Snippets.*

1. Introduction

Search engines have become the most supportive tool for obtaining useful information from the Internet. The search results obtained by most of the popular search engines are not satisfactory. It amazes users because they do input the right keywords and search engines do return pages involving these keywords, and most of the results obtained are irrelevant. Developing Web search mechanisms depends on addressing two important questions: (1) how to extract related Web pages of user interest, and (2) how to rank them according to relevance in a given set of potentially related Web pages. Measures of semantic similarity are necessary to evaluate the effectiveness of a Web search mechanism in finding and ranking results. In traditional approaches users provide manual assessments of relevance or semantic similarity. This is very difficult and expensive.

Semantic similarity between words is the study of an integral part of information retrieval and natural language processing. Semantic similarity is a concept whereby a set of terms within term lists are assigned a metric based on the likeness of their meaning. Measuring the semantic similarity between words is an important factor in different tasks on the web such as relation extraction, community mining, document clustering, automatic meta-data extraction and Web mining applications such as, community extraction, relation detection, and entity disambiguation. The main problem in information retrieval 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 numerous natural language processing jobs such as Word Sense Disambiguation (WSD), textual entailment and automatic text summarization. Semantic Similarity is the very challenging task when it comes to web, but in dictionary this problem is solved. For example, “apple” is frequently associated with computers on the Web. However, this sense of “apple” is not listed in most general-purpose dictionaries. A user, who searches for apple on the Web, may be interested in this sense of “apple” and not “apple” as a “fruit”.

Every day the new words are being added in web and the present words are given multiple meanings i.e. polysemous words. So manually maintaining these words is a very difficult task. We have proposed a Modified Pattern

Extraction Algorithm to estimate the semantic similarity between words or entities using Web search engines. Due to the massiveness of the web, it is impossible to analyze each document separately; hence Web search engines provide the perfect interface for this vast information. A web search engine gives two important information about the documents searched, Page count and Web Snippets. Page count of a query term will give an estimate of the number of documents or web pages that contain the given query term. A web snippet is one which appears below the searched documents and is a brief window of text that is searched around the query term in the document.

Page count between two objects is accepted normally as the relatedness measure between them. For example, the page count of the query *apple* AND *compute*r in Google is 977,000,000 whereas the same for *banana* AND *computer* is only 60,200,000 [as on 20 December 2012]. The more than 16 times more numerous page counts for *apple* AND *computer* indicate that *apple* is more semantically similar to *computer* than is *banana*. The drawbacks of page count is that it ignores the position of the two words that appear in the document, hence the two words may appear in the document but may not be related at all and page counts takes into account polysemous words of the query term, hence a word for example *Dhruv* will have the page counts for both *Dhruv* as the star of fortune and *Dhruv* as a name of the Helicopter.

Processing snippets is possible for measuring semantic similarity but it has the drawback of downloading a large number of web pages which consumes time, and all the search engine algorithms use a page rank algorithm, hence only the top ranked pages will have properly processed snippets. Hence there is no guarantee that all the information we need is present in the top ranked snippets.

Motivation: The search results returned by the most popular search engines are not satisfactory. Because of the vastly numerous documents and the high growth rate of the Web, it is time consuming to analyze each document separately. It is not uncommon that search engines return a lot of Web page links that have nothing to do with the user's need. Information retrieval such as search engines has the most important use of semantic similarity. It is the main problem to retrieve all the documents that are semantically related to the queried term by the user. Web search engines provide an efficient interface to this enormous information. Page counts and snippets are two useful information sources provided by most Web search engines. Hence, accurately measuring the semantic similarity between words is a very challenging job.

Contribution: We propose a Modified Pattern Extraction Algorithm to find the supervised semantic similarity measure between words by combining both page count method and web snippets method. We have used the four association measures including variants of Web Dice, Web Overlap Ratio, WebJaccard, and WebPMI to find semantic similarity between words in page count method using web search engines. The proposed approach goals to improve the correlation values, Precision, Recall, and F-measures, compared to the existing methods.

Organization: The remainder of the paper is organized as follows: Section 2 reviews the related work of the semantic similarity measures between words, Section 3 gives the problem definition, Section 4 explains the architecture of the system, and Section 5 gives the Modified Pattern Extraction Algorithm [MPEA]. The implementation and the results of the system are described in Section 6 and Conclusions are presented in Section 7.

2. Related Work

Semantic similarity between words has been important and challenging problem in data mining. Nowadays, World Wide Web (WWW) has become a vast collection of data and documents, with available information for every single user query.

Mehran Sahami et al., [1] presents a novel method for measuring the similarity between short text snippets by leveraging web search results to provide greater context for the short texts. In this paper, a similarity kernel function is defined and provided examples of its efficacy. The proposed method works on all kernel machines. A method is proposed for measuring the similarity between short text snippets that captures more of the semantic context of the snippets rather than simply measuring their term-wise similarity. Hsin-Hsi Chen et al., [2] proposed a web search with double checking model to explore the web as a live corpus. Instead of the simple web page counts and complex web page collection, the novel model, a Web Search with Double Checking (WSDC) is used to analyse snippets. They collect snippets for both the words X and Y from a Web Search Engine. They proceed by counting the occurrences of word X in the snippets for word Y and the occurrences of word Y in the snippets for word X. These values are then combined nonlinearly to compute the similarity between X and Y.

Rudi L. Cilibrasi et al., [3] has proposed the words and phrases acquire meaning from their relative semantics to other words and phrases and from the way they are used in society. It is a new concept of similarity between words and phrases based on information distance and

Kolmogorov complexity. The proposed method is applicable to all search engines and databases. This theory is then applied to construct a method to automatically extract similarity, the Google similarity distance, of words and phrases from the world-wide web using Google page counts. The main objective is to develop a new theory of semantic distance between a pair of objects is based on the background contents consisting of a database of documents. Relations between pairs of objects are extracted from the documents by just using the number of documents in which the objects occur, singly and jointly. Authors are introduced some notions underpinning the approach: Kolmogorov complexity, information distance, and compression-based similarity metric and a technical description of the Google distribution and the Normalized Google Distance (NGD). NGD uses page counts but does not take into account the context in which the words co-occur, it suffers from the drawbacks.

Dekang Lin et al., [4] proposed that, bootstrapping semantics from text is one of the greatest challenges in natural language learning. They defined a word similarity measure based on the distributional pattern of words. They demonstrate how their definition can be used to measure the similarity in a number of different domains. Many similarity measures have been proposed, such as information content, mutual information, Dice coefficient, cosine coefficient; Distance-based measurements, and feature contrast model. The similarity measure allows constructing a thesaurus using a parsed corpus. It is a new evaluation methodology for the automatically constructed thesaurus. The similarity between two objects is defined to be the amount of information contained in the commonality between the objects divided by the amount of information in the descriptions of the objects. They use a broad-coverage parser to extract dependency triples from the text corpus. Dependency triple consists of two words and the grammatical relationship between them in the input sentence. The description of a word ' w ' consists of the frequency counts of all the dependency triples that matches the pattern (w, *, *). The commonality between two words consists of the dependency triples that appear in the descriptions of both words. The main contribution of this paper is a new evaluation methodology for automatically constructed thesaurus. While previous methods rely on indirect tasks or subjective judgments, this method allows direct and objective comparison between automatically and manually constructed thesauri. Jian Pei et al., [5] proposed a projection based sequential pattern-growth approach for efficient mining of sequential patterns.

Jiang et. al., [6] combines a lexical taxonomy structure with corpus statistical information so that the semantic distance between nodes in the semantic space constructed by the taxonomy can be better quantified with the computational evidence derived from a distributional analysis of corpus data.

Philip Resnik et al., [7]-[8] presents measure of semantic similarity in an is-a taxonomy, based on the notion of information content. Experimental evaluation was performed using a large, independently constructed corpus, an independently constructed taxonomy, and previously existing and new human subject data and the results suggest that the measure performs encouragingly well and can be significantly better than the traditional edge-counting approach. Semantic similarity, as measured using information content, was shown to be useful in resolving cases of two pervasive kinds of linguistic ambiguity. In resolving coordination ambiguity, the measure was employed to capture the intuition that similarity of meaning is one indicator that two words are being conjoined. Suggestive results of a first experiment were bolstered by unequivocal results in a second study, demonstrating significant improvements over a disambiguation strategy based only on syntactic agreement. In resolving word sense ambiguity, the semantic similarity measure was used to assign confidence values to word senses of nouns within thesaurus-like groupings. A formal evaluation provided evidence that the technique can produce useful results but is better suited for semi-automated sense filtering than categorical sense selection. Application of the technique to a dictionary/thesaurus on the World Wide Web provides a demonstration of the method in action in a real-world setting.

Bollegala et al., [9] proposed a method which exploits the page counts and text snippets returned by a Web search engine. The proposed a novel approach is to compute semantic similarity using automatically extracted lexico-syntactic patterns from text snippets. They define various similarity scores for two given words P and Q, using the page counts for P, Q and P AND Q. These different similarity scores are integrated using support vector machines, to leverage a robust semantic similarity measure. Ming Li et al., [10] proposed a metric based on the non-computable notion of Kolmogorov computable distance and called it the similarity metric. General mathematical theory of similarity that uses no background knowledge or features specific to an application area. Hence it is, without changes, applicable to different areas and even to collections of objects taken from different areas.

Ann Gledson et al., [11] describes a simple web-based similarity measure which relies on page- counts only, can be utilized to measure the similarity of entire sets of words in addition to word-pairs and can use any web-service enabled search engine distributional similarity measure which uses internet search counts and extends to calculating the similarity within word-groups. The propensity of words to appear together in texts, known as their distributional similarity is an important part of Natural Language Processing (NLP). T Hughes et al., [12] proposed a method that presents the application of random walk Markov chain theory for measuring lexical semantic relatedness. Dekang Lin et al., [13] presented the information theoretic definition of similarity that is applicable as long as there is a probabilistic model. They demonstrate how their definition can be used to measure the similarity in a number of different domains. Many similarity measures have been proposed, such as information content, mutual information, Dice coefficient, cosine coefficient, Distance-based measurements, and feature contrast model. Vincent Schickel-Zuber et al., [14] present a novel approach that allows similarities to be asymmetric while still using only information contained in the structure of the ontology.

These literature surveys showed the fact that semantic similarity measures plays an important role in information retrieval, relation extraction, community mining, document clustering and automatic meta-data extraction. Thus there is need for more efficient system to find semantic similarity between words.

3. Problem Definition

Given two words X and Y, we model the problem of measuring the semantic similarity between X and Y, as a one of constructing a function semanticsim (X, Y) that returns a value in the range of 0 and 1. If X and Y are highly similar (e.g. synonyms), we expect semantic similarity value to be closer to 1, otherwise semantic similarity value to be closer to 0. We define numerous features that express the similarity between X and Y 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 (SVM) to classify synonymous and non-synonymous word pairs. Our objectives are:

i) To find the semantic similarity between two words and to increase the correlation value.

ii) To increases the Precision, Recall and the F-measure metrics of the system.

4. System Architecture

The outline of the proposed method for finding the semantic similarity using web search engine results is as shown in Fig. 1.

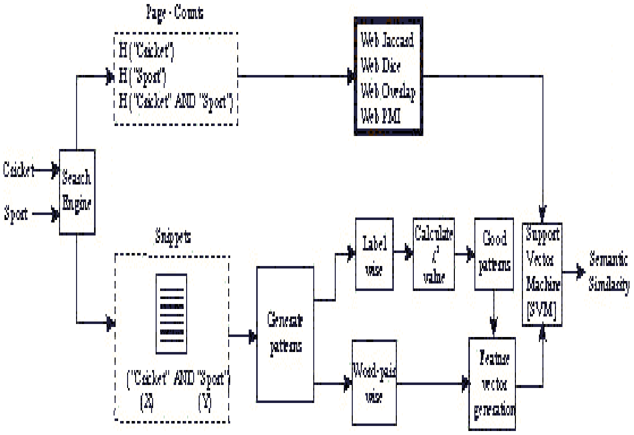

Fig. 1 System Architecture

When a query q is submitted to a search engine, web-snippets, which are brief summaries of the search results, are returned to the user. First, we need to query the word-pair in a search engine for example say we query "cricket" and "sport" in Google search engine. We get the page counts of the word-pair along with the page counts for individual words i.e. H (cricket), H(sport), H(cricket AND sport). These page counts are used to find the co-occurrence measures such as Web-Jaccard, Web-Overlap, Web-Dice and Web-PMI and store these values for future references. We collect the snippets from the web search engine results. Snippets are collected only for the query X and Y. Similarly, we collect both snippets and page counts for 200 word pairs. Now we need to extract patterns from the collected snippets using our proposed algorithm and to find the frequency of occurrence of these patterns.

We use the chi-square statistical method to find out the good patterns from the top 200 patterns of interest using the pattern frequencies. After that we integrate these top 200 patterns with the co-occurrence measures computed. If the pattern exists in the set of good patterns then we select the good pattern with the frequency of occurrence in the patterns of the word-pair else we set the frequency as 0. Hence we get a feature vector with 204 values i.e. the top 200 patterns and four co-occurrence measures values. We use a Sequential Minimal Optimization (or SMO) support vector machines (SVM) to find the optimal combination of page counts-based similarity scores and top-ranking patterns. The SVM is trained to classify the synonymous word-pairs and non-synonymous word pairs. We select synonymous word-pairs and Non-synonymous word-pairs and convert the output of SVM into a posterior probability. We define the semantic similarity between two words as the posterior probability if they belong to the synonymous-words (positive) class.

5. Algorithm

The proposed Modified Pattern Extraction Algorithm [MPEA] is used to measure the semantic similarity between words is as shown in the Table 1. Given two words A and B, we query a web search engine using the wildcard query A * * * * * B and download snippets. The * operator matches one word or none in a web page. Therefore, our wildcard query retrieves snippets in which A and B appears within a window of seven words. Because a search engine snippet contains 20 words on an 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. The algorithm which is described in the Table 1 shows that how to retrieve the patterns and the frequency of the patterns.

The Modified Pattern Extraction Algorithm as described above yields numerous unique patterns. Of those patterns only 80% of the patterns occur less than10 times. It is impossible to train a classifier with such numerous parse patterns. We must measure the confidence of each pattern as an indicator of synonymy that is, most of the patterns have frequency less than 10 so it is very difficult to find the patterns which are significant so, we have to compute their confidence so as to arrive at the significant patterns. We compute chi-square value to find the confidence of each pattern.

The chi-square value is calculated by using the formula given below:

$$\mathcal{X}^2 = \frac{(P + N)\ (p_v\ (N - n_v) - n_v(P - p_v))^2}{PN\ (p_v + n_v)(P + N - p_v - n_v)} \qquad (1)$$

Where,

P and N are the Total frequency of synonymous word pair patterns and non-synonymous word pair patterns, p_v and n_v are frequencies of the pattern v retrieved from snippets of synonymous and non-synonymous word pairs respectively.

Table 1. Modified Pattern Extraction Algorithm [MPEA]

```
Input: Given a set WS of word-pairs
Step 1:  Extract snippets using Mozbar and store it in a text file.
Step 2:  Read each snippet, remove all the non-ASCII character and
         store it in database.
Step 3:  Retrieve each snippet, check for word pair A and B, when it
         encounter, then replace word pair 'A' or 'B' by 'X' and 'Y'
         with respectively.
Step 4:  Scan the snippet until you encounter an X or Y, If you
         encounter an X go to Step 5. If you encounter a Y go to Step
         11.
Step 5:  Stop the sequence whenever you encounter a 'Y' or when the
         number of words encountered exceeds Maximum length L.
Step 6:  Scan the sequence and replace the didn't, shouldn't, etc., by
         their full forms, did not, should not etc.
Step 7:  Form the sub-sequences of the sequence such that each sub-
         sequence contains [X . . . Y . .].
Step 8:  Compare each sub-sequence with the existing patterns, If its
         unique then Add it to the list of patterns and set its count
         to 1.
Step 9:  If the sub-sequence is similar to existing pattern then frequency
         of the pattern increase by 1.
Step 10: If the length exceeds L then discard the pattern until you find
         an X or Y.
Step 11: If you encounter a Y replace the value of X with Y and Y with
         X and go to Step 5.
```

6. Implementation and Results

6.1 . Page-count-based Co-occurrence Measures

We compute four popular co-occurrence measures; Jaccard, Overlap (Simpson), Dice, and Point wise Mutual Information (PMI), to compute semantic similarity using page counts.

Web Jaccard coefficient between words (or multi-word phrases) A and B, is defined as:

$$\mathrm{WebJaccard}(A,B) = \begin{cases} 0 & \text{if } H(A \cap B) \leq c, \\ \dfrac{H(A \cap B)}{H(A) + H(B) - H(A \cap B)} & \textit{otherwise} \end{cases} \qquad (2)$$

Web Overlap is a natural modification to the Overlap (Simpson) coefficient, is defined as:

$$\mathrm{WebOverlap}(A,B) = \begin{cases} 0 & \text{if } H(A \cap B) \leq c, \\ \dfrac{H(A \cap B)}{\min(H(A), H(B))} & \text{otherwise} \end{cases} \qquad (3)$$

WebDice is defined as:

$$\text{WebDice}(A,B) = \begin{cases} 0 & \text{if } H(A \cap B) \le c, \\ \dfrac{2H(A \cap B)}{H(A)+H(B)} & \text{otherwis} \end{cases} \qquad (4)$$

Web PMI is defined as:

$$\text{WebPMI}(A,B) = \begin{cases} 0 & \text{if } H(A \cap B) \le c, \\ \log_2\left(\dfrac{\frac{H(A \cap B)}{N}}{\frac{H(A)}{N}\frac{H(B)}{N}}\right) & \textit{otherwis} \end{cases} \qquad (5)$$

We have implemented this in Java programming language and used Eclipse as an extensible open source IDE (Integrated Development Environment) [15]. We query for A AND B and collect 500 snippets for each word pair and for each pair of words (A, B) store it in the database. By using the Pattern Retrieval algorithm, we retrieved huge patterns and select only top 200 patterns. After that we compare each of the top 200 patterns based on the chi-square values "$\mathcal{X}^2$" which are called as good patterns with the patterns generated by the given word pair. If the pattern extracted for the particular word pair is one among the good patterns, store that good pattern with a unique ID and store the frequency of this pattern as that of the pattern generated by the given word pair. If a pattern does not match then store it with a unique ID and with its frequency set as 0 and store it in the table.

To the same table, we add the four co-occurrence measure values of page counts are the Web- Jaccard coefficient, Web-Overlap, Web-Dice coefficient and Web-PMI which gives a table having 204 rows of unique ID, frequency and word pair ID. After that we normalize the frequency values by dividing the value in each tuple by the sum of all the frequency values. Now this 204-dimension vector is called the feature vector for the given word pair. Convert the feature vectors of all the word-pairs into a .CSV (Comma Separated Values) file. The generated .CSV file is fed to the SVM classifier which is inbuilt in Weka software [16]. This classifies the values and gives a similarity score for the word pair in between 0 and 1.

6.2 Test Data

In order to test our system, we selected the standard Miller-Charles dataset, which is having 28 word-pairs. The proposed algorithm outperforms by 89.8 percent of correlation value, as illustrated in Table 2.

The Fig. 2 shows the comparison of correlation value of our MPEA with existing methods.

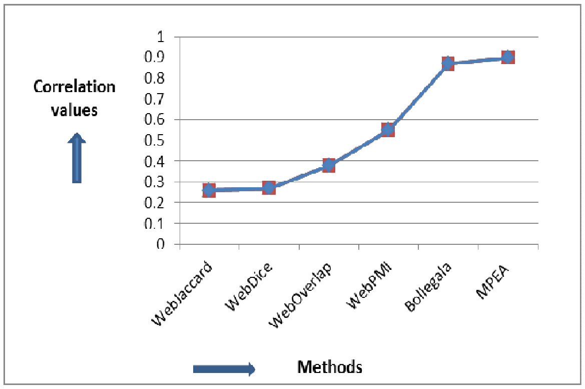

Fig. 2: Comparison of correlation value of MPEA with existing methods

The success of a search engine algorithm lies in its ability to retrieve information for a given query. There are two ways in which one might consider the return of results to be successful. Either we can obtain very accurate results or we can find many results which have some connection with the search query. In information retrieval, these are termed precision and recall, respectively [17].

The precision is the fraction of retrieved instances that are relevant, while Recall is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance. In even simpler terms, high Recall means that an algorithm returned most of the relevant results. High precision means that an algorithm returned more relevant results than irrelevant. A measure that combines both precision and recall is the harmonic mean of precision and recall which is called as the F-measure or balanced F-score.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \qquad (6)$$

Table 2. Comparison of Correlation value of MPEA with existing methods

Word Pair	**Miller-Charles**	**Web Jaccard**	**Web Dice**	**Web Overlap**	**Web PMI**	**Bollegala**	**MPEA**
automobile-car	1.00	0.65	0.66	0.83	0.43	0.92	0.98
journey-voyage	0.98	0.41	0.42	0.16	0.47	1	0.93
gem-jewel	0.98	0.29	0.3	0.07	0.69	0.82	0.98
boy-lad	0.96	0.18	0.19	0.59	0.63	0.96	0.95
coast-shore	0.94	0.78	0.79	0.51	0.56	0.97	0.98
asylum-madhouse	0.92	0.01	0.01	0.08	0.81	0.79	0.86
magician- wizard	0.89	0.29	0.03	0.37	0.86	1	0.94
midday-noon	0.87	0.1	0.1	0.12	0.59	0.99	0.71
furnace-stove	0.79	0.39	0.41	0.1	1	0.88	0.94
food-fruit	0.78	0.75	0.76	1	0.45	0.94	0.97
bird-cock	0.77	0.14	0.15	0.14	0.43	0.87	0.91
bird-crane	0.75	0.23	0.24	0.21	0.52	0.85	0.65
implement-tool	0.75	1	1	0.51	0.3	0.5	0.74
brother-monk	0.71	0.25	0.27	0.33	0.62	0.27	0.54
crane- implement	0.42	0.06	0.06	0.1	0.19	0.06	0.18
brother-lad	0.41	0.18	0.19	0.36	0.64	0.13	0.68
car-journey	0.28	0.44	0.45	0.46	0.2	0.17	0.26
monk-oracle	0.27	0	0	0	0	0.8	0.7
food-rooster	0.21	0	0	0.41	0.21	0.02	0.36
coast-hill	0.21	0.96	0.97	0.26	0.35	0.36	0.18
forest-graveyard	0.2	0.06	0.06	0.23	0.49	0.44	0.77
monk-slave	0.12	0.17	0.18	0.05	0.61	0.24	0.08
coast-forest	0.09	0.86	0.87	0.29	0.42	0.15	0.07
lad-wizard	0.09	0.06	0.07	0.05	0.43	0.23	0.03
cord-smile	0.01	0.09	0.1	0.02	0.21	0.01	0.03
glass-magician	0.01	0.11	0.11	0.4	0.6	0.05	0.04
rooster-voyage	0	0	0	0	0.23	0.05	0.06
noon-string	0	0.12	0.12	0.04	0.1	0	0.03
Correlation	**1.0**	**0.26**	**0.27**	**0.38**	**0.55**	**0.87**	**0.898**

Table 3. Precision, Recall and F-measure values for both Synonymous and Non-synonymous classes.

Class	Precision	Recall	F-Measure
Synonymous	0.9	0.947	0.923
Non-synonymous	0.83	0.714	0.769

In this paper, F-measure is computed based on the precision and recall evaluation metrics. The results are better than the previous algorithms, the Table 4 shows that the comparison of Precision, Recall and F-measure improvement of the proposed Algorithm.

Table 4. Comparison of Precision, Recall and F-measure values of MPEA with previous method

Method	Precision	Recall	F-Measure
Bollegala	0.7958	0.804	0.7897
MPEA	0.9	0.947	0.923

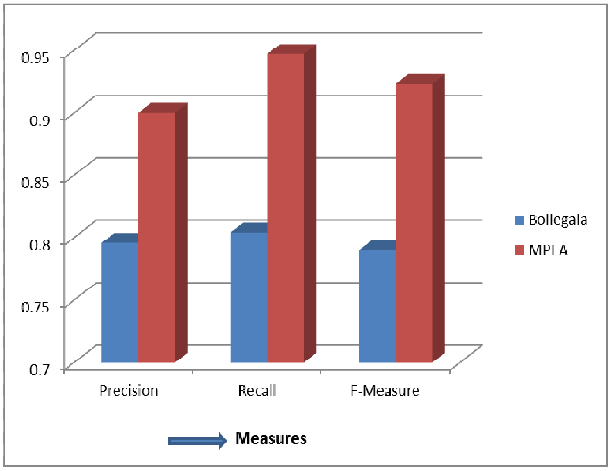

Fig. 3. Comparison of Precision, Recall and F-measure values of MPEA with previous method.

The Fig. 3 shows the comparison of Precision, Recall and F-measure values of PRA with previous method on the Miller- Charles dataset proves that our results better than the previous methods. We have tested our algorithm on the Word Similarity dataset, which contains 353 word pairs and comparison result of PRA with existing methods on the Word Similarity is as shown in the Table 5.

Table 5. Comparison result of MPEA with existing methods on the Word Similarity dataset.

Method		Correlation
Jarmasz [18]	Wordnet	0.35
Wikirelate! [19]	Wikipedia	0.48
Hughes & Ramage [20]	Wordnet	0.55
Jarmasz [18]	Roget's	0.55
Finkelstein et al. [21]	Corpus+Wordnet	0.56
Gabrilovich [22]	ODP	0.65
Gabrilovich [22]	Wikipedia	0.75
Bollegala [9]	Web snippets+ Page counts	0.74
MPEA (Proposed)	Page counts + Web snippets	0.753

7. Conclusions

Semantic Similarity measures between words plays an important role in information retrieval, natural language processing and in various tasks on the web. We have proposed a Modified Pattern Extraction algorithm to extract numerous semantic relations that exist between two words and the four word co-occurrence measures were computed using page counts. We integrate the patterns and co-occurrence measures to generate a feature vector. These feature vectors are fed to a 2- Class SVM to classify the data into synonymous and non-synonymous classes. We compute the posterior probability for each word-pair which is the similarity score for that word-pair. The proposed algorithm outperforms by 89.8 % of correlation value for Miller-Charles dataset and 75.3% of correlation value for Word similarity dataset. The Precision, Recall and F-measure values are improved compared to previous methods.

References

[1] M Sahami and T Heilman, “A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets”, 15th International Conference on World Wide Web, Year 2006, pp. 377-386.

[2] H Chen , M Lin and Y Wei , “Novel Association Measures using Web Search with Double Checking”, International Committee on Computational Linguistics, Year 2006, pp. 1009-1016.

[3] R Cilibrasi and P Vitanyi, “The Google Similarity Distance”, IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 3, Year 2007, pp. 370-383.

[4] Lin D, “Automatic Retrieval and Clustering of Similar Words”, International Committee on Computational Linguistics and the Association for Computational Linguistics, Year 1998, pp. 768-774.

[5] J Pei, J Han, B Mortazavi-Asi, J Wang, H Pinto, Q Chen, U Dayal and M Hsu, "Mining Sequential Patterns by Pattern growth: the Prefix Span Approach", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 11, Year 2004, pp. 1424-1440.

[6] Jay J Jiang and David W Conrath, "Semantic Similarity based on Corpus Statistics and Lexical Taxonomy", International Conference Research on Computational Linguistics.

[7] P Resnik, (1995) "Using Information Content to Evaluate Semantic Similarity in a Taxonomy", 14th International Joint Conference on Artificial Intelligence, Vol. 1, Year 1995, pp. 24-26.

[8] P Resnik, "Semantic Similarity in a Taxonomy: An Information based Measure and its Application to problems of Ambiguity in Natural Language", Journal of Artificial Intelligence Research, Vol. 11, Year 1999, pp. 95-130.

[9] Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka, "A Web Search Engine-based Approach to Measure Semantic Similarity between Words", IEEE Transactions on Knowledge and Data Engineering , Vol. 23, No.7, Year 2011, pp.977-990.

[10] Ming Li, Xin Chen, Xin Li, Bin Ma, Paul M and B Vitanyi, "The Similarity Metric", IEEE Transactions on Information Theory, Vol. 50, No. 12, Year 2004, pp. 3250-3264.

[11] Ann Gledson and John Keane,"Using Web Search Results to Measure Word-Group Similarity", 2nd International Conference on Computational Linguistics),Year 2008, pp. 281-288.

[12] T Hughes and D Ramage, "Lexical Semantic Relatedness with Random Graph Walks", Conference on Empirical Methods in Natural Language Processing Conference on Computational Natural Language Learning, (EMNLP-CoNLL07), Year 2007, pp. 581-589.

[13] D Lin, "An Information-Theoretic Definition of Similarity", 15th International Conference on Machine Learning, Year 1998, pp. 296-304.

[14] Schickel-Zuber V and Faltings B, (2007) "OSS: A Semantic Similarity Function Based on Hierarchical Ontologies", International Joint Conference on Artificial Intelligence, pp. 551-556.

[15] http://onjava.com/onjava/2002/12/11/eclipse.html

[16] www.cs.waikato.ac.nz/ml/**weka/**

[17] C N Pushpa, J Thriveni, K R Venugopal and L M Patnaik, "Enhancement of F-measure for Web People Search using Hashing Technique", International Journal on Information Processing (IJIP), Vol. 5, No. 4, Year 2011, pp. 35-44.

[18] M. Jarmasz, "Roget's Thesaurus as a Lexical Resource for Natural Language Processing", University of Ottowa, Technical Report, Year 2003.

[19] M. Strube and S. P. Ponzetto, "Wikirelate! Computing Semantic Relatedness using Wikipedia, "In the Proceedings of AAAI'06, Year 2006, pp. 1419–1424.

[20] T. Hughes and D. Ramage, "Lexical Semantic Relatedness with Random Graph Walks," in Proceedings of EMNLP-CoNLL'07, Year 2007, pp.581–589.

[21] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, z. Solan, G. Wolfman, and E. Ruppin, "Placing Search in Context: The Concept Revisited," ACM Transactions on Information Systems, Vol. 20,Year 2002, pp. 116–131.

[22] E. Gabrilovich and S. Markovitch "Computing Semantic Relatedness uses Wikipedia-based explicit Semantic Analysis," in Proceedings. of IJCAI'07, Year 2007, pp. 1606–1611.

Pushpa C N has completed Bachelor of Engineering in Computer Science and Engineering from Bangalore University, Master of Technology in VLSI Design and Embedded Systems from Visv esvaraya Technological University. She has 13 years of teaching experience. Presently she is working as Assistant Professor in Department of Computer Science and Engineering at UVCE, Bangalore and pursuing her Ph.D in Semantic Web.

Thriveni J has completed Bachelor of Engineering, Masters of Engineering and Doctoral Degree in Computer Science and Engineering. She has 4 years of industrial experience and 18 years of teaching experience. Currently she is an Associate Professor in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore. Her research interests include Networks, Data Mining and Biometrics

Venugopal K R is currently the Principal, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. .He received his Master's degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded Ph.D. in Economics from Bangalore University and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored 31 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++ and Digital Circuits and Systems etc. During his three decades of service at UVCE he has over 350 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining.

L M Patnaik is a Honorary Professor in Indian Institute of Science, Bangalore. During the past 35 years of his service at the Institute he has over 700 research publications in refereed International Journals and refereed International Conference Proceedings. He is a Fellow of all the four leading Science and Engineering Academies in India; Fellow of the IEEE and the Academy of Science for the Developing World. He has received twenty national and international awards; notable among them is the IEEE Technical Achievement Award for his significant contributions to High Performance Computing and Soft Computing. His areas of research interest have been Parallel and Distributed Computing, Mobile Computing, CAD for VLSI circuits, Soft Computing and Computational Neuroscience.