WEB SEARCH ENGINE BASED SEMANTIC SIMILARITY MEASURE BETWEEN WORDS USING PATTERN RETRIEVAL ALGORITHM

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.
[2]Honarary Professor, Indian Institute of Science, Bangalore.
pushpacn@gmail.com

ABSTRACT

Semantic Similarity measures plays an important 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 Pattern Retrieval Algorithm [PRA] 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 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.

KEYWORDS

Information Retrieval, Semantic Similarity, Support Vector Machine, Web Mining, Web Search Engine, Web Snippets

1. INTRODUCTION

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

The study of semantic similarity between words has been 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 component in various 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. In information retrieval, one of the main problems is to retrieve a set of documents that is semantically related to a given user query. Efficient estimation of semantic similarity between words is critical for various natural language processing tasks such as Word Sense Disambiguation (WSD), textual entailment and automatic text summarization. In dictionary the semantic similarity between words is solved, but when it comes to web, it has become the challenging task. For example, "apple" is frequently associated with computers on the Web. However, this sense of "apple" is not listed in most general-purpose thesauri or dictionaries. A user, who searches for apple on the Web, may be interested in this sense of "apple" and not "apple" as a "fruit".

As we know that new words are being added every day 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 Pattern Retrieval Algorithm to estimate the semantic similarity between words or entities using Web search engines. Due to the vastness 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 globally 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 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 vast 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 task.

Contribution: We propose a Pattern Retrieval Algorithm to find the supervised semantic similarity measure between words by combining both page count method and web snippets method. Four association measures including variants of Web Dice, Web Overlap Ratio, Web Jaccard, and WebPMI are used to find semantic similarity between words in page count method using web search engines. The proposed approach aims 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 gives the architecture of the system and Section 5 explains the proposed algorithm. 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 always been a challenging problem in data mining. Nowadays, World Wide Web (WWW) has become a huge collection of data and documents, with available information for every single user query.

Mehran Sahami et al., [1] proposes 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 method for measuring the similarity between short text snippets is proposed 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 simple web page counts and complex web page collection, the proposed novel model is a Web Search with Double Checking (WSDC) used to analyze snippets.

Rudi L. Cilibrasi et al., [3] proposed the words and phrases acquire meaning from the way they are used in society, from their relative semantics to other words and phrases. It is a new theory of similarity between words and phrases based on information distance and Kolmogorov complexity. The method is applicable to all search engines and databases. 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).

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. 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. Bollegala et al., [9] proposed a method which exploits the page counts and text snippets returned by a Web search engine. 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.

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. 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] present an information theoretic definition of similarity that is applicable as long as there is a probabilistic 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 proved 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 A and B, we model the problem of measuring the semantic similarity between A and B, as a one of constructing a function semanticsim (A, B) that returns a value in the range of 0 and 1. If A and B 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 A and B 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 improves the correlation value.

ii) To improves 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 Figure 1.

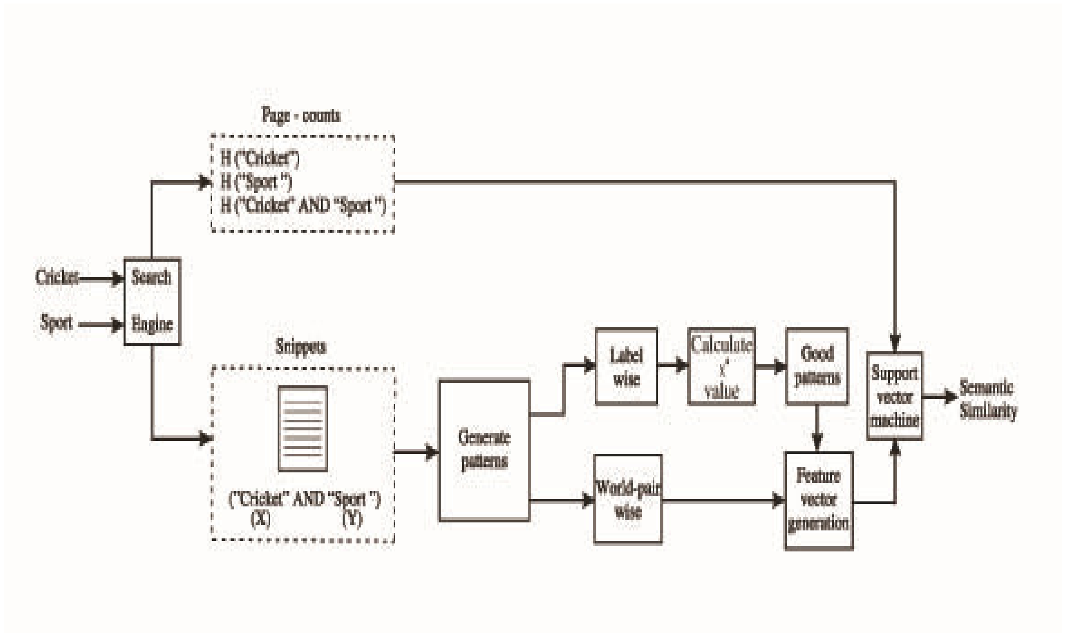

Figure. 1 System Architecture

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.

The chi-square statistical method is used 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 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 Pattern Retrieval Algorithm 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 pattern retrieval 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:

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

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. Pattern Retrieval Algorithm [PRA]

```
Input: Given a set WS of word-pairs

Step 1: Read each snippet S, remove all the non-ASCII character and store it in database.
Step 2: for each snippet S do
            if word is same as A then
                Replace A by X
            end if
            if word is same as B then
                Replace B by Y
            end if
        end for
Step 3: for each snippet S do
            if X € S then
                goto Step 4
            end if
            if Y € S then
                goto Step 9
            end if
        end for
Step 4: if Y or Number of words > Max. length L then
                stop the sequence seq.
        end if
Step 5: for each seq do
            Perform stemming operation.
        end for
Step 6: Form the sub-sequences of the sequence such that each sub-sequence
        contains [X . . . Y . .].
Step 7: for each subseq do
            if subseq is same as existing pattern and unique then
                list_ pat = list_ pat + subseq
                freq _pat = freq _pat + 1
            end if
        end for
Step 8: if length exceeds L then
            Discard the pattern until you find an X or Y.
        end if
Step 9: if you encounter Y then
            goto Step 4.

        end if
```

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:

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

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

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

WebDice is defined as:

$$\mathrm{Web\,Dice}(A,B) = \begin{cases} 0 & \text{if } H(A \cap B) \leq c, \\ \dfrac{2H(A \cap B)}{H(A) + H(B)} & \text{otherwise} \end{cases} \quad (4)$$

Web PMI is defined as:

$$\mathrm{Web\,PMI}(A,B) = \begin{cases} 0 & \text{if } H(A \cap B) \leq c, \\ \log_{2}\left(\dfrac{\frac{H(A \cap B)}{N}}{\frac{H(A)}{N}\frac{H(B)}{N}}\right) & \textit{otherwise} \end{cases} \quad (5)$$

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 "χ^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 add the four values of 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. Later 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.

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

	Web Jaccard	**Web Dice**	**Web Overlap**	**Web PMI**	**Bollegala Method**	**PRA Proposed**
Correlation Value	0.26	0.27	0.38	0.55	0.87	0.898

The Figure 2 shows the comparison of correlation value of our PRA with existing methods.

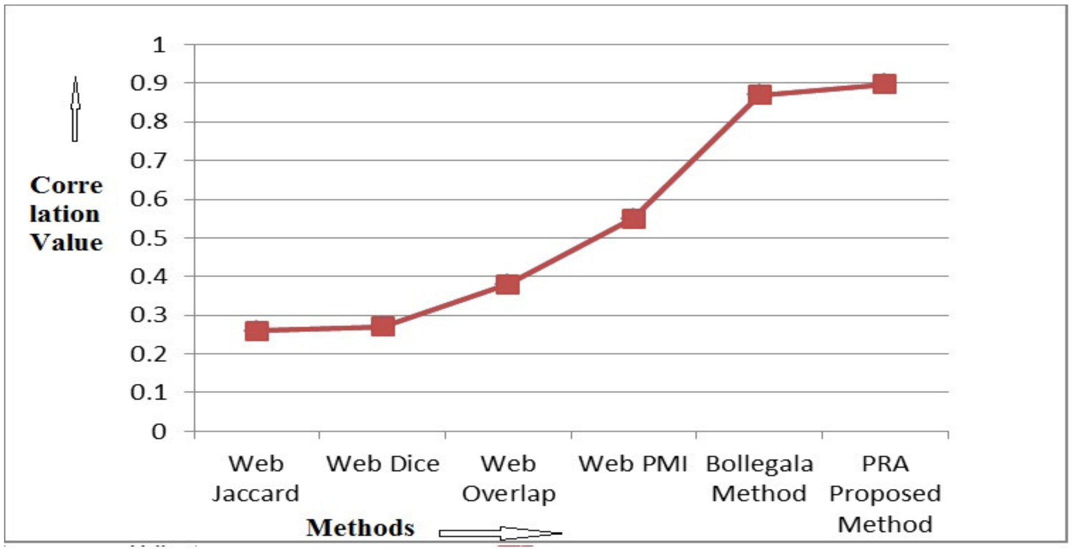

Figure 2: Comparison of correlation value of PRA with existing methods.

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 you can obtain very accurate results or you 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.

Table 2. 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	0.833	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 3 shows that the comparison of Precision, Recall and F-measure improvement of the proposed Algorithm.

Table 3. Comparison of Precision, Recall and F-measure values of PRA with previous method

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

7. CONCLUSIONS

Semantic Similarity measures between words plays an important role information retrieval, natural language processing and in various tasks on the web. We have proposed a Pattern Retrieval 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 percent of correlation value. The Precision, Recall and F-measure values are improved compared to previous methods.

REFERENCES

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

[2] Chen H, Lin M & Wei Y, (2006) " Novel Association Measures using Web Search with Double Checking", International Committee on Computational Linguistics and the Association for Computational Linguistics, pp. 1009-1016.

[3] Cilibrasi R & Vitanyi P, (2007) " The google similarity distance", IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 3, pp. 370-383.

[4] Lin D, (1998) "Automatic Retrieival and Clustering of Similar Words", International Committee on Computational Linguistics and the Association for Computational Linguistics, pp. 768-774.

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

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

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

[8] Resnik P, (1999) "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, pp. 95-130.

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

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

[11] Ann Gledson & John Keane, (2008) "Using Web-Search Results to Measure Word-Group Similarity", 22nd International Conference on Computational Linguistics), pp. 281-28.

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

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

[14] Schickel-Zuber V & 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/eclipe.html

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

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

Authors

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 Visvesvaraya 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 16 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 Masters 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 VCE he has over 250 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 Instituteof 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.