Semantic Similarity based Web Document Classification Using Support Vector Machine

Kavitha Chinniyan, Sudha Gangadharan, and Kiruthika Sabanaikam
Department of Computer Science and Engineering, PSG College of Technology, India

Abstract: *With the rapid growth of information on the World Wide Web (WWW), classification of web documents has become important for efficient information retrieval. Relevancy of information retrieved can also be improved by considering semantic relatedness between words which is a basic research area in fields of natural language processing, intelligent retrieval, document clustering and classification, word sense disambiguation etc. The web search engine based semantic relationship from huge web corpus can improve classification of documents. This paper proposes an approach for web document classification that exploits information, including both page count and snippets. To identify the semantic relations between the query words, a lexical pattern extraction algorithm is applied on snippets. A sequential pattern clustering algorithm is used to form clusters of different patterns. The page count based measures are combined with the clustered patterns to define the features extracted from the word-pairs. These features are used to train the Support Vector Machine (SVM), in order to classify the web documents. Experimental results demonstrate 5% and 9% improvement in F1 measure for Reuters 21578 and 20 Newsgroup datasets in the classifier performance.*

Keywords: *Document classification, text mining, SVM, latent semantic indexing.*

Received October 4, 2013, accepted March 19, 2014

1. Introduction

Due to the exponential growth on the Internet and the emergent need to organize them, automated categorization of documents into predefined labels has received an ever-increased attention in the recent years. Automatic document classification tasks can be divided into three types: supervised document classification where some external mechanism like human feedback provides information on the correct classification for documents, unsupervised document classification where the classification must be done entirely without reference to external information and semi-supervised document classification where parts of the documents are labeled by the external mechanism.

Classification is a form of data analysis that can be used to extract models describing important data classes. Such analysis can provide a better understanding of the data at large. Document classification can be applied as an information filtering tool and can be used to improve the retrieval results from a query process and to make good decisions. The documents to be classified may be texts, images, music etc. Each kind of document possesses its special classification problems. Documents may be classified according to their subjects or according to other attributes like document type, author, printing year etc. Mining useful information from a relatively unstructured source, such HTML, World Wide Web (WWW), news articles, digital libraries, online forums and other types of documents can be difficult. So extracting information from these resources and proper categorization and knowledge discovery is an important area for research.

Semantic similarity between terms changes over time and across domains. For example, apple is frequently associated with computers on the Web. This sense of apple is not listed in most general-purpose thesauri. A user, who searches for apple on the web, may be interested in this sense of apple and not apple as a fruit. New words are constantly being created as well as new senses are assigned to existing words. Manually maintaining thesauri to capture these new words and senses is costly if not impossible. Each source of information provides a different viewpoint; a combination has the potential of having better knowledge than any single method. Thus in our approach, the Support Vector Machine (SVM) is trained using the features extracted from the selected dataset along with the features extracted from the web to improve the classification accuracy.

The reminder of the paper is organized as follows: section 2 discusses related works, section 3 explains the existing methodology, section 4 describes the proposed approach, section 5 illustrates the proposed method with a sample set of documents, section 6 presents the experimental results and section 7 concludes the paper with future work.

2. Related Works

Peng and Choi [18], proposed to automatically classify documents based on the meanings of words and the relationships between groups of meanings or concepts.

The bag-of-words document representation is simple, yet limited with two major problems. Word count cannot differentiate between related words in different documents or same words have different meanings under different context. Thus, rather than counting word occurrences, counting word senses might improve text classification by applying semantics to classification.

Anagnostopoulos *et al*. [1] have showed how classification can be performed effectively and efficiently using a search-engine model. Elberrichi *et al*. [5] have used WordNet concept to categorize text documents but the word sense disambiguation technique is not capable of determining the correct sense of words with multiple synonyms. Gracia and Mena [7] have explored the semantic relatedness measure between two words that use web as knowledge source. Semantic relatedness measures the degree in which words or concepts are related. Latent Semantic Analysis (LSA) is a statistical technique that leverages word co-occurrence from large unlabeled corpus of texts.

In the measures based on Wikipedia, a method to represent the meaning of texts or words as weighted vectors of Wikipedia-based concepts using machine learning techniques is used. But Wikipedia is still not comparable with the whole web in the task of discovering and evaluation of implicit relationships. measures based on the web gives a guarantee of maximum coverage. Arya and Lavanya [2] have proposed a similarity measure that combines various similarity scores based on page counts and lexico-syntactic patterns extracted from text snippets. The proposed work aims to classify the web documents which are most related to user's query into predefined classes or categories. Shoham and Balabanovic [20] proposed hybrid system to deal with increasing number of users and an increasing number of documents.

Khan *et al*. [11] reviewed the different machine learning algorithms for text-document classification.

2.1. K-Nearest Neighbor

It is a classification approach [22] where objects are classified by voting several labeled training examples with their smallest distance from each object. This method is simple, non-parametric and easy to implement. But it requires more time for classifying objects for a huge training set.

2.2. Decision Trees

A decision tree classifier is a tree in which internal nodes are labeled by terms, branches departing from them are labeled by the weight and leaves are labeled by categories. It is simple to understand and interpret even for non-expert users. The major risk of implementing a decision tree is it over-fits the training data with the occurrence of an alternative tree that categorizes the training data worse.

2.3. Naive Bayes

Naive bayes classifier [17, 19] is based on Baye's theorem. It computes the posterior probability of the document and it assigns document to the class with the highest posterior probability. It requires only a small amount of training data to estimate the parameters necessary for classification. But it has a low classification performance.

2.4. Rocchio's Algorithm

The algorithm in [4] is easy to implement, efficient in computation, fast learner and have relevance feedback mechanism but low classification accuracy. The researchers have used a variation of Rocchio's algorithm in a machine learning context, i.e., for learning a user profile from unstructured text [16, 22], the goal in these applications is to automatically induce a text classifier that can distinguish between classes of documents.

2.5. Support Vector Machines

It is a supervised classification approach. SVM has the capability to handle large feature spaces. SVM was initially applied to text categorization by Joachims [9]. Joachims validated the classification performance of SVM in text categorization and it had the highest classification precision. Hence, the proposed approach uses SVM due to its effectiveness.

Pawar and Gawande [16] performed a review on different types of supervised machine learning algorithms for text classification and concluded that SVM classifier has been recognized as one of the most effective text classification method. Khan *et al*. [12], explored the main techniques and methods for automatic documents classification. In [8], it is said that there is no single representation scheme and classifier that can be recommended as a general model for any application.

3. Existing System

Bollegala *et al*. [3] have proposed an automatic method to estimate the semantic similarity between words or entities in a query using web search engine for classifying them as synonymous or non-synonymous word pairs using SVM. Given two words P and Q, the problem of measuring the semantic similarity between P and Q is modelled as a function sim(P, Q) that returns a value in range of [0, 1]. If they are highly similar, sim(P, Q) will be close to 1. On the other hand, if they are not semantically similar, then sim(P, Q) will be close to 0. There are numerous features that express the similarity between P and Q using page counts and snippets retrieved from a web search engine. Using this feature representation of words, the

SVM is trained to classify synonymous and non-synonymous word pairs.

Figure 1 illustrates an example of using the existing method to compute the semantic similarity between two words.

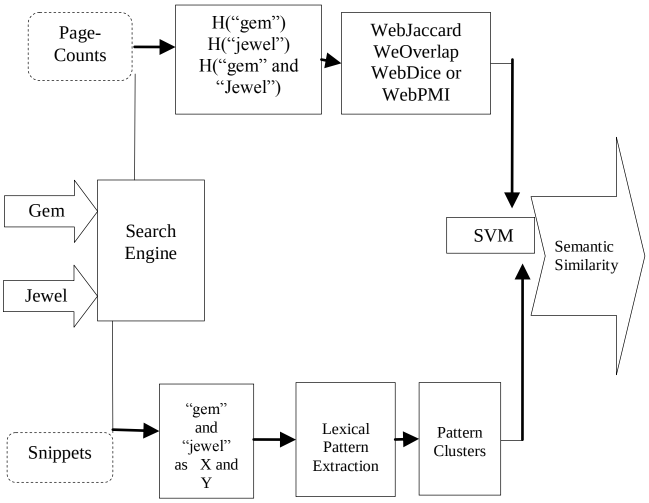

Figure 1. Outline of the existing method.

The steps are given as:

1. Query a web search engine and retrieve page counts and snippets for input word-pairs from WordNet.
2. Calculate the word co-occurrences on web documents using either of the four measures namely WebJaccard, WebDice, WebOverlap or WebPMI.
3. The frequencies of lexical patterns extracted from web snippets are calculated.
4. The lexical patterns that convey the same semantic relations are clustered together using a sequential pattern clustering algorithm.
5. Both page counts-based similarity scores and lexical pattern clusters are combined using SVM to find the semantic similarity measure.
6. The words are classified as synonymous or non-synonymous based on the similarity score.

4. Proposed System

The proposed methodology classifies the documents according to their content into certain categories. The proposed system architecture is shown in Figure 2. WordNet, a manually created English dictionary, is used to generate the training data required for the proposed method. Around 2000 nouns are randomly selected from WordNet and a pair of synonymous words from a synset of each selected noun is extracted. These word pairs are given to the search engine from which the page counts and the snippets are extracted.

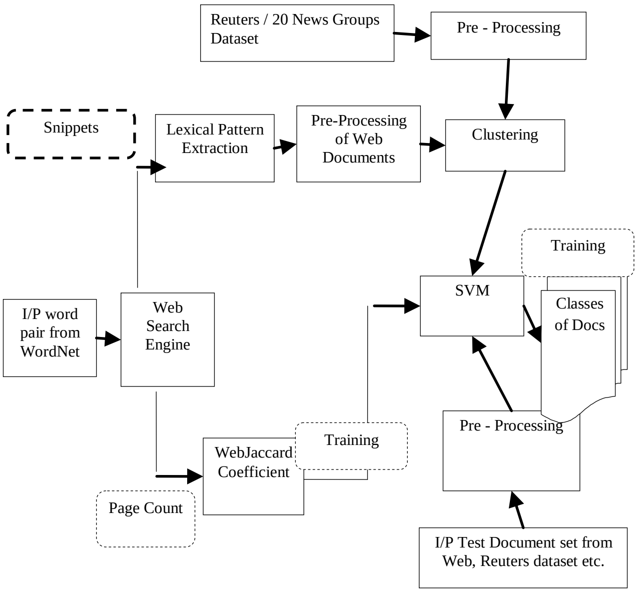
Reuters / 20 News Groups Dataset
Pre - Processing
Snippets
Lexical Pattern Extraction
Pre-Processing of Web Documents
Clustering
Training
SVM
Classes of Docs
I/P word pair from WordNet
Web Search Engine
Training
Pre - Processing
WebJaccard Coefficient
Page Count
I/P Test Document set from Web, Reuters dataset etc.

Figure 2. Outline of proposed systems.

The steps of the SVM based document classification approach are described as follows:

- *Step 1*. The WebJaccard coefficient measure for page counts is defined as:

$$WebJaccard\ (P,Q) = \frac{H(P \cap Q)}{H(P) + H(Q) - H(P \cap Q)} \qquad (1)$$

Where *P* and *Q* are two words in the query, *H*(*P*) and *H*(*Q*) denote the page counts for word *P* and *Q* respectively.

- *Step 2*. The snippets are given to the lexical pattern extraction algorithm [3] to recognize the semantic relations that exist between two words. The subsequences from the snippets are generated using the following conditions:

A subsequence must contain exactly one occurrence of each word *P* and *Q*.

1. The maximum length of a subsequence is *L* words.
2. A subsequence is allowed to skip one or more words. However, not more than *g* number of words consecutively.
3. All negation contractions must be expanded. For example, didn't is expanded to *did not*.

The frequency of occurrence of all subsequences is counted and only those sub sequences that occur more than *T* times are used as lexical patterns. The web documents corresponding to the top ranked patterns are extracted. The parameters are set experimentally to *L*= 7, *g*=2 and *T*=5.

- *Step 3*. The extracted web documents are subjected to pre-processing in order to transform the documents into a form suitable for automatic processing. The documents are represented as term vectors using Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme. The process is shown in Figure 3.

The TF-IDF is a well known approach to compute the term weights to ensure the effectiveness of document classification. The weight of term *i* in document *j* is given by:

$$tf * idf_{i,j} = tf_{i,j} * idf_i \tag{2}$$

Where term frequency is calculated as:

$$tf_{i,j} = \frac{N_{i,j}}{NT_j} \tag{3}$$

Where $N_{i,j}$ is the number of times the term *i* appears in the document *j* and NT_j is the total number of terms in the document *j*. The inverse document frequency is calculated as:

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

Where |*D*| is the total number of documents and |*d*: t_i *ε d*| is the number of documents in which the term t_i appears. These TF-IDF values and the list of documents are then formed as a vector space. The feature selection is usually employed to reduce the size of the feature space to an acceptable level in order to increase the overall performance.

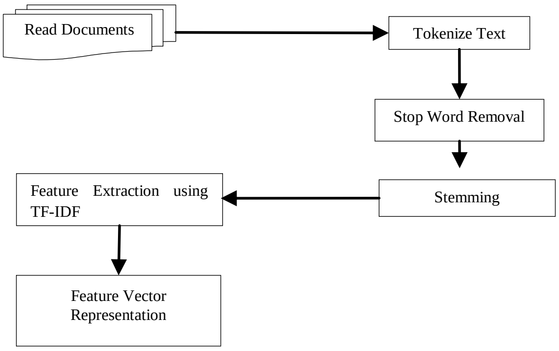
Read Documents
Tokenize Text
Stop Word Removal
Stemming
Feature Extraction using TF-IDF
Feature Vector Representation

Figure 3. Feature selection process.

- *Step 4*. Cluster of similar documents are formed and labeled using Latent Semantic Indexing (LSI) which analyzes the relationship between a set of documents and it uses Singular Value Decomposition (SVD) to find the semantic similarity between documents [10, 14]. LSI constructs a term-document matrix, A, to identify the m unique terms within a collection of n documents where each term is represented by a row and each document is represented by a column with each matrix cell initially representing the number of times the associated term appears in the indicated document. SVD is performed on the matrix [14, 23] to determine patterns in the relationships between the terms and concepts contained in the document. It computes the term and document vector spaces using the relation:

$$A = TSD^T \tag{5}$$

Where *T*=*m* by *r* term concept vector matrix; *S*=*r* by *r* singular value matrix; *D*=*n* by *r* concept document vector matrix and *r*=rank of *A*. LSI modifies the SVD to reduce the rank of *S* to size *k*, which effectively reduces the size of term and document vector matrix. This SVD reduction preserves the most important semantic information in the document and ignores the noise and other undesirable influences. This reduced set of matrices is denoted with a modified formula such as:

$$A » A_k = T_k S_k D_k^T \tag{6}$$

The similarity of terms and documents within these vector spaces shows how close they are to each other. It is computed as a function of the angle between the corresponding vectors as:

$$cos\,\theta = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} * \sqrt{\sum_{i=1}^{n}(B_i^2)}} \tag{7}$$

- *Step 5*. The page counts-based co-occurrence measures and the snippets-based lexical pattern clusters are combined into one feature vector and are used to train the SVM. Training is the process of taking the content that is known to belong to specified classes and creating a classifier on the basis of that known content [21]. SVM is trained using the existing training set of Reuter's dataset [13] and web documents retrieved from top ranked snippets. The groups of training documents are classified into different classes based on the cosine similarity measure as in Equation 7. The LibSVM[1] is used as the SVM implementation. As the LibSVM[1] can be used for multiclass classification [15], it can be applied for any number of classes. Finally, the results are analyzed by running the classifier on other contents and labeling them as belonging to one class.

5. Illustration

Initially the word-pair "gold and silver" is taken from WordNet. The top ranked snippets related to the word-pair is extracted using the lexical pattern extraction algorithm. The proposed work can be illustrated using three documents D1, D2 and D3 which were retrieved from the snippets.

- *D1*: Shipment of gold damaged in afire.
- *D2*: Delivery of silver arrived in a silver truck.
- *D3*: Shipment of gold arrived in a truck.

Tokenizing the Terms:

T1: A T2: Arrived T3: Damaged
T4: Delivery T5: Fire T6: Gold

T7: In	T8: Of	T9: Shipment
T10: Silver	T11: Truck	

Once the terms are extracted, the term-document matrix (A) is constructed by computing the weights using a specific term weight scoring system like TF-IDF as in Table 1.

Table 1. Term-document matrix A.

Term/Document	D1	D2	D3
T1	1	1	1
T2	0	1	1
T3	1	0	0
T4	0	1	0
T5	1	0	0
T6	1	0	1
T7	1	1	1
T8	1	1	1
T9	1	0	1
T10	0	2	0
T11	0	1	1

The term document matrix is decomposed into three new matrices according to Equation 5. The SVD results after the calculation are shown in Tables 2 and 3.

Table 2. Term vector coordinates (T).

Term/Document	D1	D2	D3
T1	-0.4201	0.0748	-0.0460
T2	-0.2995	-0.2001	0.4078
T3	-0.1206	0.2749	-0.4538
T4	-0.1576	-0.3046	-0.2006
T5	-0.1206	0.2749	-0.4538
T6	-0.2626	0.3794	0.1547
T7	-0.4201	0.0748	-0.0460
T8	-0.4201	0.0748	-0.0460
T9	-0.2626	0.3794	0.1547
T10	-0.3157	-0.6093	-0.4013
T11	-0.2995	-0.2001	0.4078

Table 3. Document vector coordinates (D^T).

D1	D2	D3
-0.4945	0.6458	-0.5817
-0.6492	-0.7194	-0.2496
-0.5780	0.2556	0.7750

The values of singular value matrix (S) are 4.0989, 2.3616 and 1.2737.The dimensionality is reduced by 'k = 2' values according to Equation 6. The reduced term-vector matrix and reduced document vector matrix are given in Tables 4 and 5.

Table 4. Reduced term vector matrix (T_k).

Term/Document	D1	D2
T1	-0.4201	0.0748
T2	-0.2995	-0.2001
T3	-0.1206	0.2749
T4	-0.1576	-0.3046
T5	-0.1206	0.2749
T6	-0.2626	0.3794
T7	-0.4201	0.0748
T8	-0.4201	0.0748
T9	-0.2626	0.3794
T10	-0.3157	-0.6093
T11	-0.2995	-0.2001

Table 5. Reduced document vector matrix ($D^T{}_k$).

D1	D2	D3
-0.4945	0.6458	-0.5817
-0.6492	-0.7194	-0.2496

The values of reduced singular value matrix (S_k) are 4.0989 and 2.3616.

The cosine similarity is applied to the training documents to form the clusters. The page counts for the word-pair is taken and the WebJaccard coefficient is calculated as [0.0542]. This page count measure is combined with the snippets-based clusters to train the SVM. Similarly the training documents of Reuter's dataset are pre-processed and their clusters are used to train the SVM. During the training of SVM with Reuter's training set alone, the two classes namely "gold" and "coffee" were not identified. But when the SVM was trained with a combination of both Reuter's training set and web documents retrieved from the queried word-pairs, these two classes were identified, which indicates a thorough exploration of concepts.

A test document (t) "Gold and Silver arrived in truck." is given to the classifier. The vector coordinates for the test document is [-0.2140, -0.1821]. The cosine similarity between the test document and the three documents are calculated as: Sim(t, D1) = -0.0541, Sim(t, D2) = 0.9910, Sim(t, D3) = 0.4478

With these values, the given test document is more similar to D2 and it is clustered with that document D2. The training documents are used in this way to form different class labels by forming clusters of similar documents. Thus, LSI recovers the original semantic structure of the space and its original dimensions.

6. Experimental Results

The major goal of document classification is to classify the documents relevant to user query. For the experiment, 2000 word pairs were taken from WordNet. Numerous patterns were extracted from the snippets.

Table 6. Comparison of classification using Reuter's training set and a combination of Reuter's and web documents.

Class	SVM Trained with Web Document and Reuters		SVM Trained with Reuters Training Set Only	
	Precision	Recall	Precision	Recall
Earn	0.87	0.82	0.82	0.77
Acquisition	0.85	0.85	0.85	0.75
Money	0.89	0.92	0.76	0.80
Grain	0.93	0.95	0.88	0.82
Crude Oil	0.86	0.88	0.87	0.84
Trade	0.87	0.83	0.81	0.78
Interest	0.90	0.85	0.83	0.81
Ship	0.83	0.88	0.76	0.81
Gold	0.88	0.84	-	-
Coffee	0.86	0.88	-	-
No.of Input Documents	50	50	50	50
Average	0.88	0.87	0.82	0.80

The web documents were retrieved for the patterns and clustered into many categories which were used for training the SVM. 50 input test documents from REUTERS 21578 dataset was given for testing. It was classified into ten largest classes as per the corpus REUTERS 21578 as in Table 6.

The SVM was trained separately with the Reuters dataset also. While testing the system with the same 50 input test documents, only eight classes were obtained. As per the results shown in Table 6, the categories namely coffee and gold were not identified when the SVM was trained with the Reuter's dataset. There was a percentage increase in the accuracy of classification as the coverage of data was more in-depth. The experiment was repeated by giving 200 and 350 test documents as input to the system in order to evaluate the performance of classification accuracy as shown in Figure 4.

The standard performance measure for document classification is Precision, Recall and F1-Measure [6]. While comparing the classification of documents based on web documents and classification based on Reuters dataset, the results based on a combination of web documents and Reuters gave more new categories to be included and also the precision and recall gave a performance increase. But the Reuters dataset was classified into only one of the predefined categories.

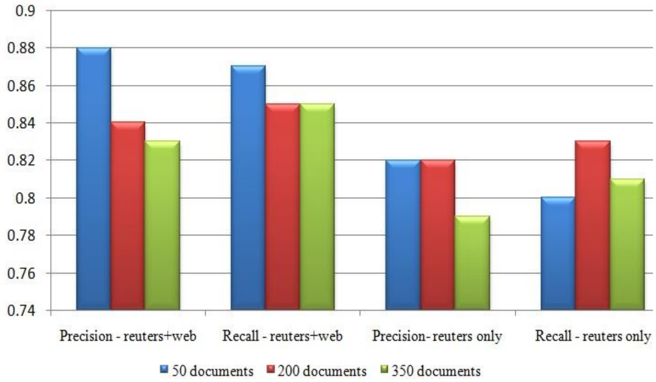

Figure 4. Classification using Reuter's dataset and a combination of Reuter's dataset and web documents.

The SVM was again trained with another dataset named 20News Groups. The description of the different categories formed with 20News Group dataset is shown in Table 7.

Table 7. Description of categories for 20 news groups.

Categories	Description
Motorcycle	Motorcycle and Autos
Hardware	Windows and MAC
Sports	Baseball and Hockey
Graphics	Computer graphics
Religion	Christianity, Atheism and misc

The experiment was repeated by giving 50, 200 and 350 input test documents for testing the performance of classification accuracy. The precision and recall for 20 News group dataset is shown in Figure 5.

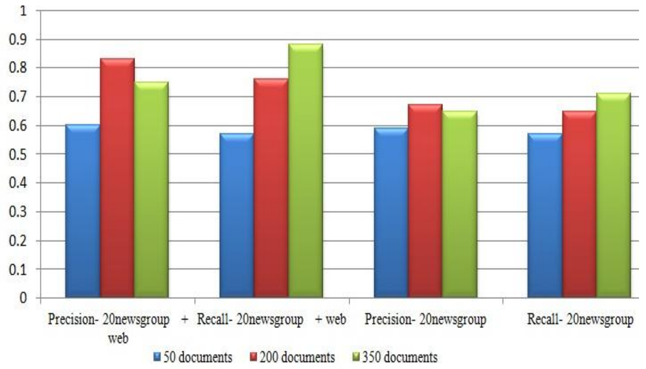

Figure 5. Classification using 20 news groups dataset and a combination of 20 news groups dataset and web documents.

The macro F_1 measure is calculated using Equation 8.

$$F_1 = 2 * \frac{Precision\ .\ Recall}{Precision + Recall} \tag{8}$$

Table 8 shows the performance comparison of F_1 measure for the data sets Reuter's and 20 News group.

Table 8. Macro F_1 measure.

Data set	Average Precision	Average Recall	F1 Measure
Using Reuter's and Web Features	.85	.86	86%
Using Reuters Dataset Features Alone	.81	.81	81%
Using 20 News Group and Web Features	.73	.74	73%
Using 20News Group Dataset Features Alone	.64	.64	64%

7. Conclusions and Future Work

Document classification is processed using SVM and the semantics obtained from extracting the snippets and page counts from the web search engine. Training set is derived by using both the web search engine semantic and concept-based extraction using LSI in order to retain the semantics among documents. A comparison of training the SVM using Reuter's dataset alone and combination of web documents and Reuter's dataset has been carried out. The $F1$ measure of classification based on the proposed methodology is 86% and $F1$ measure for classification based training using Reuter's dataset alone is 81%. The $F1$ measure of classification based on the proposed methodology in 20Newsgroup dataset is 73% and $F1$ measure for classification based on training 20 news group dataset alone is 64%. The experimental results indicate that the proposed method based on web documents yield better performance on unstructured documents due to the dynamic update of web contents and a thorough exploration of concepts.

The future work can include the classification of documents based on evolutionary techniques. An evolutionary algorithm deploys a randomized search. It is capable of searching through very complex problem spaces and get good results quickly for problems that change over time. In order to reduce the processing time the clustering phase can be parallelized.

Acknowledgment

Our thanks to Dr. R Rudramoorthy, Principal, PSG College of Technology, Dr. R.Venkatesan, Professor and Head, Department of CSE and Mr K Chidambaram Kollengode, Director, Cloud and Big Data Analysis, Nokia R&D, Bangalore, India, for their support. This work is done to support PSG-Nokia research on big data Analytics.

References

[1] Anagnostopoulos A., Broder A., and Punera K., "Effective and Efficient Classification on a Search-Engine Model," *in Proceeding of the 15th ACM International Conference on Information and Knowledge Management*, Virginia, pp. 1-29, 2007.

[2] Arya S. and Lavanya S., "An Approach for Measuring Semantic Similarity between Words Using SVM and LS-SVM," *in Proceeding of International Conference on Computer Communication and Informatics*, Coimbatore, pp. 1-4, 2012.

[3] Bollegala D., Matsuo Y., and Ishizuka M., "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, 2011.

[4] Cohen W. and Singer Y., "Context-Sensitive Learning Method for Text Categorization," *ACM Transactions on Information Systems*, vol. 17, no. 2, pp. 141-173, 1999.

[5] Elberrichi Z. and Rahmoun A., Mohd.Amine .B, "Using WordNet for Text Categorization," *The International Arab Journal of Information Technology*, vol. 5, no. 1, pp. 16-24, 2008.

[6] Forman G. and Scholz M., "Apples-To-Apples in Cross-Validation Studies: Pitfalls in Classifier Performance Measurement," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, pp. 49-57, 2010.

[7] Gracia J. and Mena E., "Web-Based Measure of Semantic Relatedness," *in Proceeding of 9th International Conference on Web Information Systems Engineering*, Auckland, pp. 136-150, 2008.

[8] Harish B., Guru D., and Manjunath S., "Representation and Classification of Text Documents: A Brief Review," *International Journal of Computer Application*, no. Special Issue, pp. 110-119, 2010.

[9] Joachims T., "Text Categorization with Support Vector Machines Learning with Many Relevant Features," *in Proceeding of the 10th European Conference on Machine Learning*, Chemnitz, pp. 137-142, 1998.

[10] Kavitha C., Sadasivam G., and Priya M., "Annotation-Based Document Classification Using Shuffled Frog Leaping Algorithm," *International Journal of Computational Science and Engineering*, vol. 9, no. 3, pp. 215-221, 2014.

[11] Khan A., Baharudin B., Lee L., and Khan K., "A Review of Machine Learning Algorithms for Text-Documents Classification," *Journal of Advances in Information Technology*, vol. 1, no. 1, pp. 4-20, 2010.

[12] Khan A., Bahaurdin B., and Khan K., "An Overview of E-Documents Classification," *in Proceeding of International Conference of Machine Learning and Computing*, Perth, pp. 544-552, 2009.

[13] Lewis D., "Reuters-21578 Text Categorization Collection," *University of California*, 1997.

[14] Muflikhah L. and Baharudin B., "High Performance in Minimizing of Term-Document Matrix Representation for Document Clustering," *in Proceeding of International Conference on Innovative Technologies in Intelligent systems and Industrial Applications*, Kuala Lumpur, pp. 225-229, 2009.

[15] Pant B. and Mayor S., "Document Classification Using Support Vector Machine," *International Journal of Engineering Science and Technology*, vol. 4, no. 4, pp. 1741-1745, 2012.

[16] Pawar P. and Gawande S., "A Comparative Study on Different Types of Approaches to Text Categorization," *International Journal of Machine Learning and Computing*, vol. 2, no. 4, pp. 423-426, 2012.

[17] Pazzani M. and Billsus D., "Learning and Revising User Profiles, The Identification of Interesting Web Sites," *Machine Learning*, vol. 27, no. 3, pp. 313-331, 1997.

[18] Peng X. and Choi B., "Documents Classification Based on Word Semantic Hierachies," *in Proceeding of the International Conference on Artificial Intelligence and Applications*, Innsbruck, pp. 362-367, 2005.

[19] Rish I., "An Empirical Study of the Naïve Bayes Classifier," *in Proceeding of the IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, pp. 41-46, 2001.

[20] Shoham Y. and Balabanovic M., "Content-based Collaborative Recommendation," *Communications of the Association for Computing Machinery*, vol. 40, no. 3, pp. 66-72, 1997.

[21] Sun A., Lim E., and Liu Y., "On Strategies for Imbalanced Text Classification Using SVM: A Comparative Study," *Decision Support Systems*, vol. 48, no. 1, pp. 191-201, 2009.

[22] Tam V., Santoso A., and Setiono R., "A Comparative Study of Centroid-Based,

Neighborhood-Based and Statistical Approaches for Effective Document Categorization," *in Proceeding of the 16th International Conference on Pattern Recognition*, Quebec, pp. 235-238, 2002.

[23] Yang J. and Watada J., "Decomposition of Term-Document Matrix Representation for Clustering Analysis," *in Proceeding of International Conference of Fuzzy Systems*, Taipei, pp.976-983, 2011.

Kavitha Chinniyan is working as an Assistant Professor (Senior Grade) in Department of Computer Science and Engineering in PSG College of Technology, India. She is pursuing her research work in Semantics in Large Scale Distributed Systems. Her area of interests includes semantic web technology, parallel processing and data structures. She has published 5 papers in referred Journals and 4 papers in Conferences.

Sudha Gangadharan is working as a professor in CSE Department of PSG College of Technology. She has 20 years of teaching experience. Her area of interest includes distributed systems and software engineering. She has published 5 books, 30 papers in referred Journals and 32 papers in National and International Conferences. She has coordinated two AICTE-RPS projects in the areas of distributed computing. She is the coordinator of PSG-Yahoo research in grid and cloud computing, Nokia Research on Big Data Analytics and Xurmo Research in social networking.

Kiruthika Sabanaikam is a Post Graduate student of ME-Software Engineering in Department of Computer Science and Engineering in PSG College of Technology, India. Her area of interest is data mining and semantic web technology.