A Comparative Study of Machine Learning Approaches- SVM and LS-SVM using a Web Search Engine Based Application

S. S. Arya
P.G.Scholar,
Department of Computer Science and Engineering,
Sona College of Technology,
Salem, India
aryadeep88@gmail.com

S.Lavanya
Asst. Professor,
Department of Computer Science and Engineering,
Sona College of Technology,
Salem, India
lavansadeesh@yahoo.com

Abstract— **Semantic similarity refers to the concept by which a set of documents or words within the documents are assigned a weight based on their meaning. The accurate measurement of such similarity plays important roles in Natural language Processing and Information Retrieval tasks such as Query Expansion and Word Sense Disambiguation. Page counts and snippets retrieved by the search engines help to measure the semantic similarity between two words. Different similarity scores are calculated for the queried conjunctive word. Lexical pattern extraction algorithm identifies the patterns from the snippets. Two machine learning approaches- Support Vector Machine and Latent Structural Support Vector Machine are used for measuring semantic similarity between two words by combining the similarity scores from page counts and cluster of patterns retrieved from the snippets. A comparative study is made between the similarity results from both the machines. SVM classifies between synonymous and non-synonymous words using maximum marginal hyper plane. LS-SVM shows a much more accurate result by considering the latent values in the dataset.**

Keywords- **Web Mining, Support Vector Machine, Latent Support Vector Machine**

Introduction

Information available on the web considered as vast, hidden network of classes of objects, are interconnected by various semantic relations. Semantics identify concepts which allow extraction of information from data. The measurement of semantic similarity between words remains a challenging task in many Natural Language Processing tasks and information retrieval such as Query Expansion, Query suggestion, Word Sense Disambiguation etc. The semantic similarity between entities changes over time and domain. General purpose ontology such as WordNet [6] is not efficient since manually maintaining thesauri is impossible as sense of a word changes.

An automatic method of estimating semantic similarity using search engines is more efficient [5]. Page counts and snippets are the two types of useful information provided by a search engine. Page count of a query is the number of web pages returned by the query concerned. It is the number of hits as more words are appended to a query string using the AND operator. Page counts for two words are considered as the global co-occurrences of those words on the web. Two words are considered to be more similar if they have more page count. For example page count for query apple and computer in Google is 288,0000 and that for banana and computer is 359,0000[5].It indicates that apple is more semantic related to computer than banana. Snippet is the text document that shows the sample content to users about the web search engine result page. It provides a

convenient summary about search results. So it obviates the need to download the source documents from the web. Snippets for a query containing two words capture the local context. Consider a snippet from Google for the query New Delhi and India.

"New Delhi is the most expensive city in India for expatriates in terms of cost of living"

Here the phrase 'is the most expensive' shows a hypernymic relation with New Delhi and India. Several patterns are identified using such relations from snippets. The pattern is formed by replacing the queried words by variables X and Y. An automatically extracted lexical syntactic patterns approach is used to compute the similarity between words using text snippets retrieved from search engine. A lexical pattern extraction algorithm generates sub sequences from snippets. Since a semantic relation can be expressed using more than one pattern, clustering the patterns showing the same semantic relation helps to represent the relation between words accurately. Support Vector Machine (SVM) [8] combines the co-occurrences measures calculated from page counts and those lexical patterns from snippets and to find the similarity measure.

In this paper, we use two machine learning approaches called support vector machine (svm) and latent structural support vector machines (ls-svm) [7] to compute the similarity measure. Since svm cannot handle the missing data which occur frequently in statistical data analysis [13], the proposed system makes use of ls-svm which makes use of latent variables. The proposed system compares the similarity scores from both svm and ls-svm. Svm classifies between synonymous words and non-synonymous words by creating a maximum marginal hyper plane.

I. RELATED WORK

The semantic similarity between words, when the knowledge base considered as graph, uses a metric called distance [1]. The shortest path in a is-a hierarchy is used to measure the distance between concepts. The conceptual distance between two words which are represented by two nodes in an is-a semantic net is the minimum number of edges separating the nodes. The problem with this approach is that it considers a uniform distance for all links in taxonomy.

Besides the natural way to evaluate semantic similarity by considering distance, an alternative way to measure similarity is based on the notion of information content [2]. Similarity of two concepts is based on the extent to which they share common information. The upper bound of two concepts is considered as a highly specific content. Word similarity is also taken into account in case of multiple inheritance. The widely acknowledged problem with this approach is that word sense disambiguation is not considered. So it produces similarity measure for words on the basis of irrelevant word senses.

Another approach for measuring semantic similarity takes in account of multiple information sources [3]. The similarity measure in a lexical taxonomy which is structured in a tree like hierarchy with a node for a concept has irregularities in their link density. So in addition to the minimum path length, the local semantic density and the depth of the subsumer in the hierarchy is also considered into account. Depth is measured by counting the level from subsumers to the top of hierarchy. Semantic density is considered as the information content of a concept which is calculated from a large corpus. Path length and semantic density is calculated from a lexical database.

Double checking method uses text snippets returned by a search engine to measure semantic similarity between words [4]. Two objects are considered to be associated if one can be found out from the other one by web search. For the two words X and Y they collect snippets for each word from a web search engine. The occurrences of word X in snippets for word Y and the occurrences of word Y in the snippets for word X are noted. These values are combined to compute the similarity between the words. The major problem with this approach is that we cannot assure the occurrence of one word in the snippet for the other even though they are related.

An approach for measuring semantic similarity takes into account of snippets alone [15]. The snippets are downloaded from the Wikipedia and are preprocessed for stop word removal and stemming. Porter algorithm is used for stem removal. Similarity measures are based on five co-occurrence measures such as simple matching, Dice, Jaccard, Overlap and cosine coefficient. A widely acknowledged problem with this approach is that snippets are mainly dependent on the ranking algorithm used by the search engines. The resulting web pages may not be relevant to the user query.

Support Vector Machines are well suited for text classification [17].The goal of text classification is to categorize the documents into a fixed number of predefined categories. The accurate classification is obtained by using SVM. The first step in text categorization is to transform the documents which are a string of characters into an attribute value representation. Results show that SVM shows accurate classification over the

existing methods such as Rocchio's algorithm, Bayes classification-Nearest Neighbor algorithm, C4.5 algorithm. SVMs do not require manual parameter tuning.

A web search engine based approach [5] for measuring semantic similarity is used in query expansion, word sense disambiguation [11]. The idea of calculating the semantic similarity between words using web search engines is via page counts and snippets. Various similarity scores are calculated from page counts and lexico syntactic patterns [9] are extracted from snippets. These different similarity scores are integrated using support vector machines to measure semantic similarity. But the major problem is that SVM [13] cannot handle missing data which occur frequently in statistical data analysis.

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the US-letter paper size. If you are using A4-sized paper, please close this file and download the file for "MSW A4 format".

II. METHODOLOGY

A. Outline

The proposed system exploits page counts and snippets retrieved from search engines. Page counts are retrieved for the individual words and their conjunctive. Various co-occurrences measures such as WebJaccard, WedDice, WebOverlap, and WebPMI are calculated from page counts [5]. Snippets are also extracted from search engine for the conjunctive query which represents the local context in which two words co-occur on the web. The numerous semantic relations are identified from snippets. Numerous lexical syntactic patterns are identified and the frequency of each pattern is calculated. The lexical patterns that are conveying the similar semantic relations are clustered together to effectively represent the semantic relation between words. The clustering of snippet patterns is done using the pattern clustering algorithm. Both page count based similarity scores and clusters of lexical patterns define numerous features that identify relation between words. Then both SVM and LS-SVM are trained to measure the semantic similarity. Once trained, the machines can be used for measuring similarity between different word pairs.

Page counts are more closely related with the actual word co-occurrences in the web. Page counts actually represent the global co-occurrence of two words on the web, while the snippets show the local context of query word. It is shown that there is a high correlation between word counts obtained from a search engine and that from a corpus such as brown corpus [12]. The combination of page count and snippets give more accurate result than by using the individual concepts.A shallow lexical pattern extraction algorithm which is a modified version of prefix span algorithm [9] generates sub sequences from the snippets. When given a snippet retrieved for a word pair,

i. Replace the two words with variables.
ii. Replace all numeric values by marker.

Generate sub sequences of words which satisfy the following conditions:

i. Subsequence must contain exactly one occurrence of each variable.
ii. Maximum length should be L words.
iii. Subsequence can be formed by skipping one or more words. But the total length of gaps should not be more than G words.
iv. Expand all the negations.

Count the frequency of all sub sequences and select the sub sequences which occur more than N times.

A semantic relation can be expressed using more than one pattern. By identifying such patterns that convey the similar relation helps to represent the relation between words accurately. A lexical pattern clustering algorithm is used to cluster together such patterns. A machine learning approach combines together the similarity score from page counts and the pattern clusters from the snippets. SVM and LS-SVM are used for classification of synonymous and non-synonymous words. But the major problem with the SVM approach is that it cannot handle the missing data which occur frequently in the statistical data analysis. So Latent Structural SVM (LS-SVM) provides better classification by allowing latent variables.

B. System Architecture

In order to present the complete idea, we depict the system architecture of the proposed system in Figure 1.

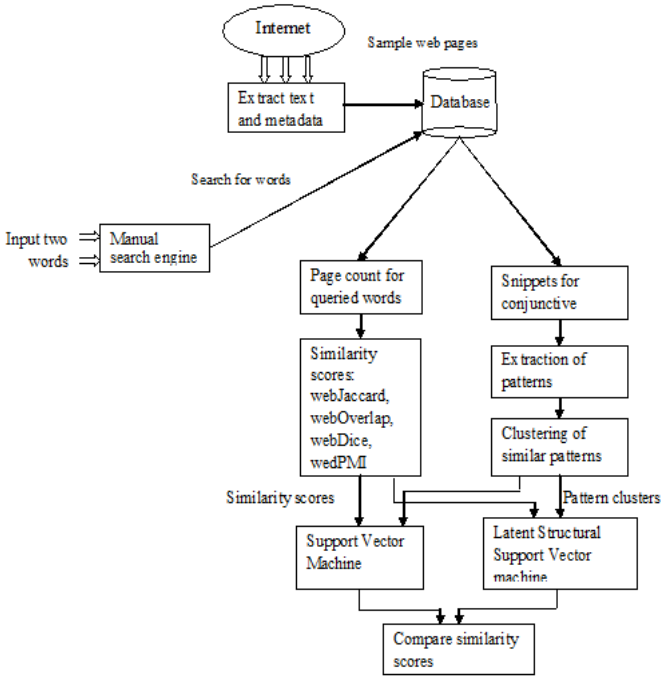

Fig 1: Architecture of the proposed system

Sample pages are collected from the Google search engine. These pages are pre-processed by converting the many forms that the words occur into suitable index terms. Index terms are more suitable for content representation in the web pages. The text and meta data are collected from the pages and are stored into the database. When the user searches for the words in the interface, the control goes to the database. Page counts along with the similarity scores such as webJaccard, webDice, WebOverlap, WebPMI are calculated and retrieved from the database. Meanwhile the database also retrieves the snippets for the conjunctive query. Since the snippets contain the summary of contents and url of different web pages, the user can identify the relevant information for the requested query. Relevant patterns are identified for the requested query using a pattern extraction algorithm which is a modified version of prefix span algorithm. The patterns showing similar semantic relations are clustered together using pattern clustering algorithm. The similarity scores along with the pattern clusters are represented as feature vectors and are given to both SVM and LS-SVM. The results are the analyzed.

C. Machine Learning Approaches

The machine learning approaches SVM and LS-SVM have their own advantages. The SVM technique is independent of dimensionality of the feature vector space. It normally uses linear model to create a non-linear output space representation and an input space is transformed into a high dimensional space. It can handle input space of any dimensionality by providing over fitting protection. The technique is independent of the number of input features. SVM uses maximum margin hyperplane to classify between two class separable datasets. In our experiments the words are selected from search engine by taking into account of page counts and snippets. Both the machines are tested and trained using words from the search engines after some pre-processing steps. Thus obtained training words are classified into two classes- synonymous and non-synonymous.

When given a training vector $x_i \in R^n$ in two classes and a labeled vector $y \in R^l$, SVM solves the optimization problem [8] as

$$\begin{aligned} &\min_w \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t.\, y_i(x_i^T w + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0 \end{aligned} \qquad (1)$$

In case of SVM technique, there should be maximum distance from the hyperplane to the data points. The data points which are nearer to the hyperplane shows almost incorrect decision. Such points can be grouped into either of the two classes. The main aim of SVM is to find out the maximum margin hyperplane that divides the

points belonging to two different classes. SVM learns the parameters w and b on the training sample datasets which actually represent the hyperplane that best separates the data points. By maximizing the margin we get the safest classification of classes. The margin is maximized by minimizing the value of (||ω||^2)/2 in (1). The tuning parameter C is the capacity that weighs the degree of misclassification. C should be low so that the margin is wider. Figure 2. shows the working of a two class SVM using support vectors and maximum margin hyperplane.

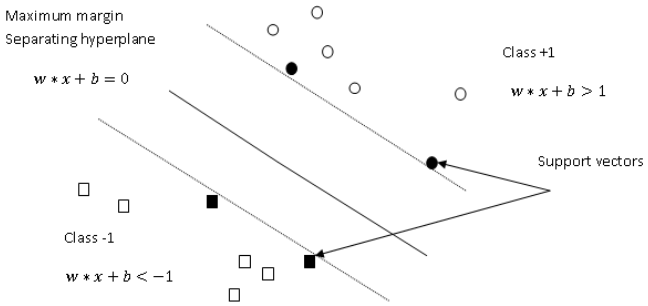

Fig 2: Maximum margin hyperplane and support vectors for a two class SVM

In machine learning, the combination of hidden variables and the observed variables offer more expressive power than using observed variables alone. SVM cannot express the latent or missing values. LS-SVM can be used in cases where a hidden structure is useful in a classification technique. In many applications in natural language processing, some of the most important data may be missing in the training data set, yet it is necessary for a successful classification technique. The latent variables have no role in the output, but they are just intermediate representations. In this paper, synonymous words from Google search engine are collected and given as training data into both the machines. The missing data from the data set are handled by the LS-SVM to classify between synonymous and non-synonymous words. The latent variables in this work actually represent the rank value of the web pages. The input corresponds to the words from web pages after pre processing, output corresponds to the classification of synonymous and non-synonymous words and the latent variables or hidden variables corresponds to the rank of web pages.

The structural SVM allows much more feasible feature construction than SVM technique by providing a control for over fitting by maximizing the margin between the outputs. The parameters of latent variables can be learned from the given training dataset. In SVM input-output pair is characterized as, (x,y)εX x Y , pairs in the training dataset, but the structural SVM also depends upon the set of unobserved latent variables, hεH. A feature vector that shows the relation between input x, output y, latent variable h is represented as (x,y,h) while in SVM it is (x,y).

LS-SVM can handle the latent variables by solving an associated non-convex optimization problem [7] as,

$$\min_{w} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\left(\max_{(\hat{y},\hat{h})\in Y\times H}[\vec{w}\cdot\phi(x_i,\hat{y},\hat{h}) + \Delta(y_i,\hat{y})]\right) - C\sum_{i=1}^{n}\max_{h\in H}[\vec{w}\cdot\phi(x_i,y_i,h)] \quad (2)$$

The loss function Δ in (2) measure how the predicted structured output differs from correct output. The loss function is independent of the latent variables since in many real world applications, the hidden data may not be a part of the output but may be some intermediate representations. The associated non-convex optimization problem can be solved by many algorithms. Convex-Concave procedure [14] is the widely used one.

Once the machine gets trained it can be used to compute the similarity between words. The proposed system also makes use of a comparative study between the similarity measures from both SVM and LS-SVM.

III. DATA ANALYSIS AND RESULTS

For the experiments in the proposed work, we use information from the Google search engine. Information available from the search engines come in many forms. Some sample HTML pages are collected from the web by querying the word pairs. These pages are pre-processed by converting the many forms that the words occur into suitable index terms. Index terms are more suitable for content representation in the web pages. The text and meta data are collected and stored into the database. In our experiments we use MySQL database. An interface is created for the search engine where the users can search for the stored words in the database. This interface is created using PHP, a server side scripting language. The PHP code is interpreted by the XAMP server which makes use of MySQL.

A number of web pages are selected according to the ranks assigned by the search engine. The top ten results of a queried word show the most relevant results. If a word pair is queried into the search engine, the retrieved results are stored in database according to their rank value. In our experiments we use word pairs such as gem-jewel, cricket-sports, car-automobile etc which are taken from the Miller-Charles dataset. The web pages containing the queried conjunctive words are stored according to the relevance factor. When the queried word is given to the interface it returns the page count for the individual words as well as for the conjunctive word. Generally, if we are giving words P and Q, then page counts are retrieved for the words P, Q, P AND Q. Then page count co-occurrence measures [5] such as webJaccard, webOverlap, webDice, WebPMI are calculated to evaluate the similarity between words. The snippets which are the actual content representation of web pages are also retrieved from the database. By analyzing the snippets, the most relevant web pages can be selected by clicking the URL associated with it. Several patterns are extracted from the snippets using pattern extraction algorithm. By analyzing the patterns we can identify the semantic relation between the words clearly. The patterns that show the same semantic relations are clustered together using the pattern clustering algorithm.

SVM and LS-SVM are used for combining the page count based co-occurrence measures and cluster of patterns. The web pages having high ranks are considered to be the most relevant result for the queried words. Those pages are most likely to contain the exact synonymous words. For instance if two words such as apple and computer are queried the top ranking results shows the relation of apple and computer while the less relevant pages may contain the information regarding apple as a fruit. So depending on the ranks the words are classified. The web pages that are considered to be more relevant contain the exact synonymous words. Fig.3.shows the result obtained from SVM which classifies synonymous and non-synonymous words.

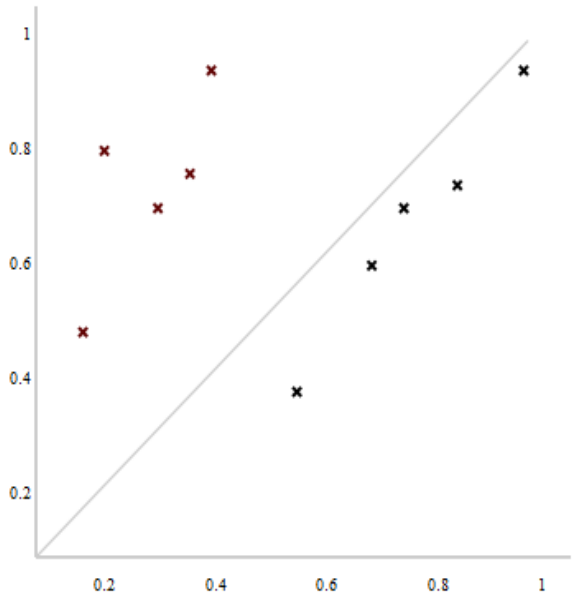

Fig 3: SVM classification result

The separating margin hyperplane differentiates between words. The data points on the right side of the separating hyperplane shows the synonymous word pairs present in the sample web pages while those on the left side shows the non-synonymous word pairs. The data points which are farther away from the separating plane are considered to be accurate. In our experiments, the similarity measure ranges in the interval [0, 1]. The graph obtained above can be converted into similarity scores by using sigmoid function.

The words from the search engine are then trained using LS-SVM. LS-SVM takes into account of hidden values also. The hidden values do not have any role in the classification result. But they play an important role in the accuracy of classification. The minutest ranking results are also considered as latent variables. Thus LS-SVM shows the exact classification of synonymous and non-synonymous word pairs. Fig. 4 shows the classification result of LS-SVM which gives an accurate classification by considering more data points which are having less rank values.

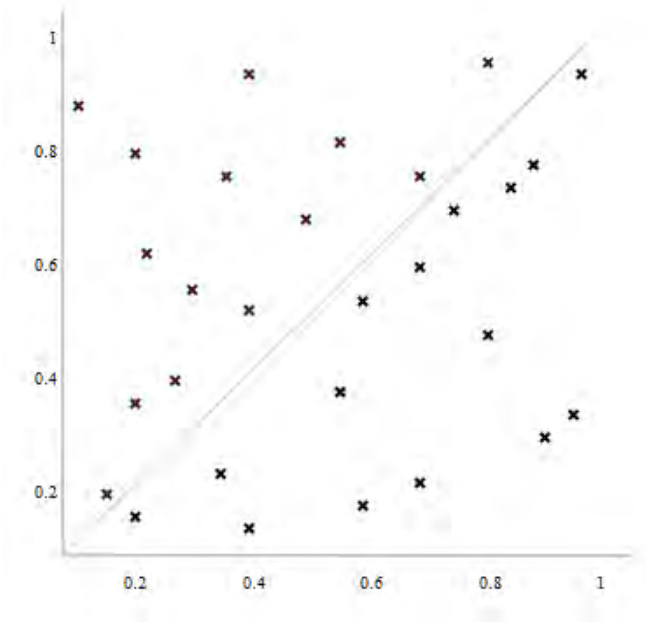

Fig 4: LS-SVM classification

The data points on the right side of the separating margin plane show synonymous word pairs and those on the left side show the non-synonymous words. In LS-SVM the input is a collection of words, output is a two class classification of synonymous and non-synonymous words. The latent variables correspond to the minutest ranks in the web pages. When both the results are compared, LS-SVM shows the accurate classification of a web page. By analyzing the result, the web pages containing the maximum synonymous word pairs can be identified.

IV. CONCLUSION

We proposed the comparative analysis of two machine learning approaches- SVM and LS-SVM using an application which is based on web search engines. The application uses page counts and snippets. The page count based co-occurrence measures and pattern clusters from snippets are combined and are represented as feature vectors. They are used for testing and training SVM and LS-SVM. The LS-SVM combines both observed variables and latent variables. In our experiments, the hidden variable corresponds to the minutest rank values of the web pages. The result shows that LS-SVM provides more accurate classification between synonymous and non-synonymous words than the SVM.

REFERENCES

[1] R. Rada, H.Mili, E.Bichnell and M.Blettner, “Development and Application of a Metric on Semantic Nets”, IEEE Trans. Systems, Man and Cybernetics, 1989, vol. 19, no. 1, pp. 17-30.
[2] P.Resnik, “Using Information Content to Evaluate Semantic Similarity in taxonomy”, Proc. 14th Int’l Joint Conf. Artificial Intelligence ,1995.
[3] D.Mclean, Li.Y and Z.A.Bandar, “An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources”, IEEE Trans. Knowledge and Data Eng., 2003, vol. 15, no. 4, pp. 871-882.
[4] H.Chen, M.Lin and Y.Wei, “Novel Association Measures Using Web Search with Double Checking” , Proc. 21st Int’l Conf. Computational Linguistics and 44th Ann. Meeting of the Assoc. for Computational Linguistics (COLING/ACL ’06), 2006, pp. 1009-1016.
[5] Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka, “A Web Search Engine-Based Approach to Measure Semantic Similarity between Words” IEEE ,2011.
[6] George A. Miller , “WordNet: A Lexical Database for English”.
[7] Chun-Nam John Yu and Thorsten Joachims “Learning Structural SVMs with Latent Variables”.
[8] V.Vapnik, “Statistical Learning Theory”. Wiley, 1998.

[9] J.Pei , J.Han, B.Mortazavi-Asi,J.Wang, H.Pinto, Q.Chen, D.Dayal, and M.Hsu, " Mining Sequential Patterns by Pattern-Growth: The Prefix span Approach", IEEE Trans. Knowledge and Data Eng., 2004,vol. 16, no. 11, pp. 1424-1440.
[10] A.Bagga and B.Baldwin, "Entity-Based Cross Document Coreferencing Using the Vector Space Model", Proc. 36th Ann Meeting of the Assoc. for Computational Linguistics and 17th Int'l Conf. Computational Linguistics (COLING-ACL), 1998, pp. 79-85 .
[11] P.Resnik, "Semantic Similarity in a Taxonomy: An Information Based Measure and Its Application to Problems of Ambiguity in Natural Language", J. Artificial Intelligence Research,1999, vol. 11, pp. 95-130 .
[12] F.Keller, M.Lapata, "Using the Web to Obtain Frequencies for Unseen Bigrams", Computational Linguistics, 2003,vol. 29, no. 3, pp. 459- 484.
[13] K.Pelckmans, J.A.K Suykens and B.De Moor, "Handling Missing Values in Support Vector Machine Classifiers", Proc IJCNN IEEE 2005.
[14] AL Yullie and A.Rangarajan, "The Concave-Convex Procedure" ,2003.
[15] A.Sheetal, Takale and Sushma S. Nandgaonkar: "Measuring Semantic Similarity between Words Using Web Documents", IJACSA International Journal of Advanced Computer Science and Applications,2010.
[16] Thorsten Joachims , "Text Categorization with Support Vector Machines: Learning with Many Relevant Features".