

OPTIMIZATION OF WORD SENSE DISAMBIGUATION USING CLUSTERING IN WEKA

Neetu Sharma, Dr. S. Niranjana

¹Computer Science & Engg., Mewar University, Chittorgarh, Rajasthan, India, ²Principal,PDM college of Engg., Bahadurgarh, Haryana, India

E-Mail: neetush75@gmail.com, niranjan.hig41@gmail.com

ABSTRACT

In the Natural Language Processing (NLP) community, Word Sense Disambiguation (WSD) has been described as the task which selects the appropriate meaning (sense) to a given word in a text or discourse where this meaning is distinguishable from other senses potentially attributable to that word. These senses could be seen as the target labels of a classification problem. Clustering and classification are two important techniques of data mining. Classification is a supervised learning problem of assigning an object to one of several pre-defined categories based upon the attributes of the object. While, clustering is an unsupervised learning problem that group objects based upon distance or similarity. Each group is known as a cluster. In this paper we make use of data file poach.arff containing 7 attributes and 37 instances to perform an integration of clustering and classification techniques of data mining. We compared results of simple classification technique (using Random Forest classifier) with the results of integration of clustering and classification technique, based upon various parameters using WEKA (Waikato Environment for Knowledge Analysis), a Data Mining tool. The results of the experiment show that integration of clustering and classification gives promising results with utmost accuracy rate and robustness.

Keywords: machine learning software, data mining, data preprocessing, data visualization, WEKA, WORDNET, K-Means, Random Forest.

1. Introduction

The information age has been characterized by the development and convergence of computing, telecommunications and multilingual information systems. This has resulted in the availability of enormous volumes of information in electronic media, but whose natural language form, unlike the data presentation formats typical of computer systems in the past, is more suited for human users than computer systems. This has prompted the development of technologies that would solve this problem and support faster and more efficient access to this information. Natural Language Processing (NLP) provides tools and techniques that facilitate the implementation of natural language-based interfaces to computer systems, enabling communication in natural languages between man and machine. These techniques also enable people to organize, extract and use the knowledge contained in these huge collections of natural language electronic data. Examples of Language Technology (LT) applications include

Machine Translation (MT), Information Extraction (IE), Information Retrieval (IR), document classification and summarization, speech recognition and synthesis, to name a few.

However, a pervasive problem afflicting most LT applications is that of ambiguity. Many words have more than one meaning, depending on the context of use. The process by which the most appropriate meaning of an occurrence of an ambiguous word is determined is known as Word Sense Disambiguation (WSD), and remains an open problem in NLP. For humans, resolving ambiguity is a routine task that hardly requires conscious effort. In addition to having a deep understanding of language and its use, humans possess a broad and conscious understanding of the real world, and this equips them with the knowledge that is relevant to make sense disambiguation decisions effortlessly, in most cases. However, creating extensive knowledge-bases which can be used by computers to 'under-stand' the world and reason about word meanings accordingly, is still an unaccomplished

goal of Artificial Intelligence (AI). Consequently, approaches to automatic WSD mainly focus on knowledge-lean methods.

1.1 Machine learning algorithms:

Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too large to be covered by the set of observed examples (training data). Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new cases.

1.2 WEKA:

WEKA,[20] formally called Waikato Environment for Knowledge Learning, is a computer program that was developed at the University of Waikato in New Zealand for the purpose of identifying information from raw data gathered from agricultural domains. WEKA supports many different standard data mining tasks such as data preprocessing, classification, clustering, regression, visualization and feature selection. The basic premise of the application is to utilize a computer application that can be trained to perform machine learning capabilities and to derive useful information in the form of trends and patterns. WEKA is an open source application that is freely available under the GNU general public license agreement. Originally written in C, the WEKA application has been completely rewritten in Java and is compatible with almost every computing platform. It is user friendly with a graphical interface that allows for quick set up and operation. WEKA operates on the predication that the user data is available as a flat file or relation. This means that each data object is described by a fixed number of attributes that usually are of a specific type, normal alpha-numeric or numeric values. The WEKA application allows novice users a tool to identify hidden information from database and file systems with simple to use options and visual interfaces.

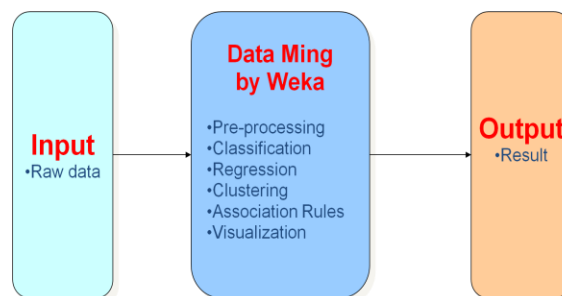


Figure 1. Working of WEKA

The WEKA system is not so much a single program as a collection of inter dependent programs bound together by a common user interface. Typically these modules fall into three categories: data set processing, machine learning schemes, and output processing. The processing of data sets involves extracting information about a data set for the user, splitting data sets into test and training sets, filtering out features in the data not required by the user, and translating the data set into a form suitable for a machine learning scheme to work with. Machine learning schemes are implementations of machine learning algorithms and typically take a converted data set and produce some output, normally a rule et. Output processing modules are concerned with taking the output from a machine learning scheme performing some task with it, such as evaluating a rule set against a test file or displaying the output in a window for the user.

Clustering

The Cluster tab opens the process that is used to identify commonalties or clusters of occurrences within the data set and produce information for the user to analyze. There are a few options within the cluster window that are similar to those described in the Classify tab. These options are: use training set, supplied test set and percentage split. The fourth option is classes to cluster evaluation, which compares how well the data compares with a pre-assigned class within the data. While in cluster mode, users have the option of ignoring some of the attributes from the data set. This can be useful if there are specific attributes causing the results to be out of range, or for large data sets. Figure 6 shows the Cluster window and some of its options.

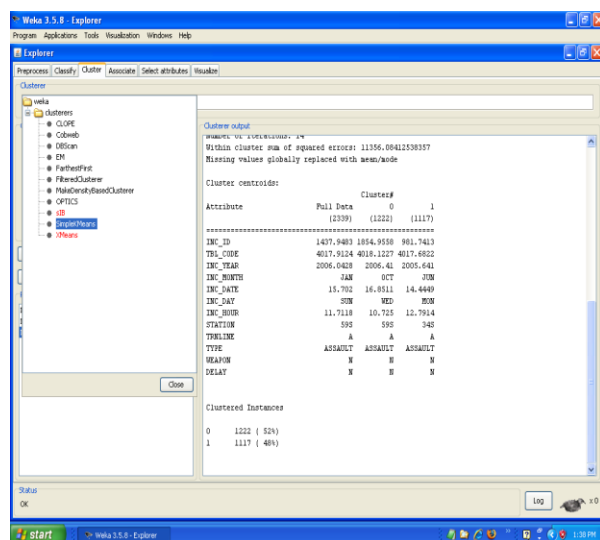


Figure 2. Clustering options

1.3 Word Sense Disambiguation

The task of WSD is a historical one in the field of Natural Language Processing (NLP). WSD was first formulated as a distinct computational task during the early days of machine translation in the late 1940s, making it one of the oldest problems in computational linguistics. Weaver (1949) introduced the problem in his now famous memorandum on machine translation:

If one examines the words in a book, one at a time through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of words. "Fast" may mean "rapid"; or it may mean "motionless"; and there is no way of telling which. But, if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then, if N is large enough one can unambiguously decide the meaning.

At that time, researchers had already in mind essential ingredients of WSD, such as the context in which a target word occurs, statistical information about words and senses, knowledge resources, etc. Very soon it became clear that WSD was a very difficult problem, also given the limited means available for computation. The 1950s then saw much work in estimating the degree of ambiguity in texts and bilingual dictionaries, and applying simple statistical models. WSD mainly has two approaches:

Knowledge Based Approaches

These approaches are mainly using external lexical resources such as dictionaries, thesaurus, WordNet etc. these are easy to implement because they require simple look up of a knowledge resources like a

machine readable dictionary. Here no need of a corpus-tagged or untagged, since no training is involved. So many algorithms are suggested with these. Some of the algorithms are Lesk Algorithm, Walkers Algorithm etc..

Machine Learning Approaches

In machine learning approaches, systems are trained to perform the task of word sense disambiguation. In these approaches, what is learned is a classifier that can be used to assign as yet unseen examples to one of a fixed number of senses. These approaches vary as the nature of the training material, how much material is need, the degree of human intervention, the kind of linguistic knowledge used, and the output produced. But the system accuracy can definitely be improved by machine learning methods. These approaches can be mainly classified into two.

Supervised Learning

In such approaches, a learning system is presented with a training set consisting of feature encoded inputs along with their appropriate label, or category. The output of the system is a classifier system capable of assigning labels to new feature encoded inputs. Here a disambiguated corpus is available for training. There is a training set of exemplars where each occurrence of the ambiguous word 'w' is annotated with a semantic label. The task is to build a classifier which correctly classifies new cases based on their context of use. Two of the supervised algorithms applied to WSD in statistical language processing is:

Unsupervised Learning

In unsupervised learning we don't know the classification of the data in the training sample. It can often be viewed as a clustering task. Hyper lex and Lin's Approach are the main two algorithms used in these techniques. Several disambiguation systems have been developed for various languages like English, Tamil, Malayalam, Hindi, Chinese etc. So many approaches are implemented in both knowledge based and machine learning methods. Hybrid approaches by combining multiple knowledge sources and using tagged data are also one of the approaches to WSD.

2. Problem Statement

The problem in particular is a comparative study of classification technique algorithm Random Forest with an integration of Simple KMeans clusterer and Random Forest classifier on various parameters using poach..arff data file containing 7 attributes and 37 instances..

3. Proposed Work

Classification is the process of finding a set of models that describe and distinguish data classes and concepts, for the purpose of being able to use the model to predict the class whose label is unknown. Clustering is different from classification as it builds the classes (which are not known in advance) based upon similarity between object features.. Integration of clustering and classification technique is useful even when the dataset contains missing values. In this experiment, object corresponds to poach.arff file from Wordnet and has two object class labels corresponds to data file. Apply classification technique using Random Forest classifier in WEKA tool. Classification is a two step process, first, it build classification model using training data. Every object of the dataset must be pre-classified i.e. its class label must be known, second the model generated in the preceding step is tested by assigning class labels to data objects in a test dataset

The test data may be different from the training data. Every element of the test data is also classified in advance. The accuracy of the classification model is determined by comparing true class labels in the testing set with those assigned by the model. Apply clustering technique on the original data file using WEKA tool and now we are come up with a number of clusters. It also adds an attribute cluster to the data set. Apply classification technique on the clustering result data set. Then compare the results of simple classification and an integration of clustering and classification. In this paper, we identified the finest classification rules through experimental study for the task of using Weka data mining tool.

4. Information Sources

4.1 WORDNET:

WordNet[12] is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets. WordNet was developed by the Cognitive Science Laboratory, at Princeton University under the direction of Professor George A. Miller The main difference between Wordnet and a dictionary is that, in Wordnet, the lexicon is divided into five categories: nouns, adjectives, verbs, adverbs and function nouns. It resembles a thesaurus because it attempts to organize lexical information in terms of word meanings, rather than word forms. Word meanings are grouped into synsets. A semantic relation is a relation between meanings. Thus a semantic relation can be

viewed as pointers between synsets. Wordnet is basically organized by semantic relations. Some of the examples of semantic relations are as explained below:

_ Synonymy: Two words are said to be synonyms, if they are similar in meaning.

_ Antonymy: Two words are said to be antonyms, if they are opposite in meaning.

_ Hyponymy: It is a semantic relation between word meanings. A word 'x' is said to be a hyponym of 'y', if 'x' is a 'kind of 'y'. For example, maple is a hyponym of tree and tree is a hyponym of plant. A hyponym inherits all the features of the more generic concept and adds at least one feature, which distinguishes it from the generic concept.

_ Meronymy: A word 'x' is said to be a meronym of 'y', if 'y' has a 'y' or 'x' is a part of 'y'. For example, leg is a hyponym of table, implying leg is a part of table or table has a leg.

_ Morphological relations: They are lexical relations between word forms. Inflectional morphological relations, is one of the types of different morphological relations. The word 'trees' is an inflectional morphological form of the word 'tree'.

Word sense disambiguation techniques,[6] both supervised and unsupervised, sometimes use only the presence or absence of words surrounding the ambiguous word as input information. This is called a bag-of-words approach. In this case, information about the co-occurrence of the ambiguous word with others is used to determine its correct sense. However, quite often, an external source of information is used to provide more advanced features, such as part-of-speech of the ambiguous word itself or surrounding words. A popular information source for general text is WordNet, a general-English lexical resource [7]. It is frequently used for both its semantic and syntactic information to disambiguate words in general texts

5. Building Classifiers

Random Forest

Random Forest is an implementation of trees structure that builds decision trees from a set of training data in the same way as J48, using the concept of Information Entropy. . The training data is a set $S = s_1, s_2, \dots$ of already classified samples.

Each sample $s_i = x_1, x_2, \dots$ is a vector where x_1, x_2, \dots represent

attributes or features of the sample. Trees are efficient to use and display good accuracy for large amount of data. At each node of the tree, the classifier chooses one attribute of the data that most

effectively splits its set of samples into subsets enriched in one class or the other.

K-Means clusterer

Simple K-Means is one of the simplest clustering algorithms K-Means algorithm is a classical clustering method that group large datasets in to clusters. The procedure follows a simple way to classify a given data set through a certain number of clusters. It select k points as initial centroids and find K clusters by assigning data instances to nearest centroids. Distance measure used to find centroids is Euclidean distance.

6. Implementation

We have chosen the data from WORDNET which have the word with the correct part of speech. For each one, we have created a line of data specifying the values for the chosen features and also the correct meaning (all separated by commas). One possibility is to enter this data into a spreadsheet (one feature per column) and then export it using "Save As" with file type "Text CSV" (set the field delimiter to be a comma). For instance, given the above features, the entry for the word "poach" appearing in the context:

Everyone I could see wanted to poach the best graduates. would be:

to, 1, steal

Non-numerical values can be put in double quotes ("...") and indeed it can be done for unusual characters also.

Weka needs a file called XXX.arff in order to build a classifier (set of rules). Here XXX can be any name .The file we have just created is in the correct format for the second part of an "XXX.arff" file for Weka The first part of the file is used to describe the format of data. This contains, after a name for the "relation" that is represented in the file, for each feature ("attribute") in turn (in the same order as in the data file), a specification of the possible values of the feature. For the above example, the file in arff format is:

```
@relation poach
@attribute word-1 string
@attribute the5 numeric
@attribute meaning { steal,boil}
@data
```

This is then followed by the lines of data. This example shows the 3 types of features most likely to have - those with string values, those with numerical values and those with a small number of possible values. Now Weka is used to build a classifier for this data file poach.arff. The steps are as follows:

1. Run Weka by selecting "Applications->Programming->Weka...". Select the Explorer
2. Click "Open file..." and select your ARFF file
3. Most classification algorithms need to know, for string-valued features, all the possible values they can have.. In the Preprocess tab of Weka, we have done the following for each string-valued feature:
 1. Click "Choose" under "Filter" and select the filter called filters->unsupervised->attribute->StringToNominal. (You don't need to do this after you have used this filter once).
 2. When the name appears after the "Choose" button, click on the name and in the box that appears enter the position (e.g. 1 or 2) of the feature we want to process after "attributeIndex".
 3. Click "OK" and the box disappears.
 4. Click "Apply".
4. Go to the Classify tab.
5. Choose a classifier in the same way as chosen a filter before.
6. The attribute that is to be predicted ("meaning" in the example above; you might have given it a different name) appears in the box below the "Test options" area.
7. Click "Start".
8. Various information about what has been learned and how well it accounts for the data have been provided. For the example "poach" data, the rules are:
 9. (word-2 = nil) => meaning=boil (8.0/0.0)
 10. (word+2 = in) => meaning=boil (2.0/0.0)
 11. => meaning=steal (22.0/6.0)
 i.e. if the word two before "poach" is empty (there is no such word), then the meaning is "boil"; otherwise if the word two after "poach" is "in" then the meaning is "boil"; otherwise the meaning is "steal".

7. Results and Conclusions

This paper presents a comparison between supervised Machine Learning Algorithms Random Forest and combination of unsupervised machine learning algorithm K-Means clusterer and Random Forest Machine learning algorithm. When applied for Word Sense Disambiguation we worked with above said two algorithms and concluded that using clustering before classification on data file poach.arff from WORDNET has optimized the performance. The dataset file poach.arff is used as input to WEKA for

the above said algorithms. On the contrary, Word Sense Disambiguation available datasets are represented as matrices of many columns (attributes) and usually few rows (examples) with lots of zeros (sparse information). Both the concepts and the representation appear very sparsely.

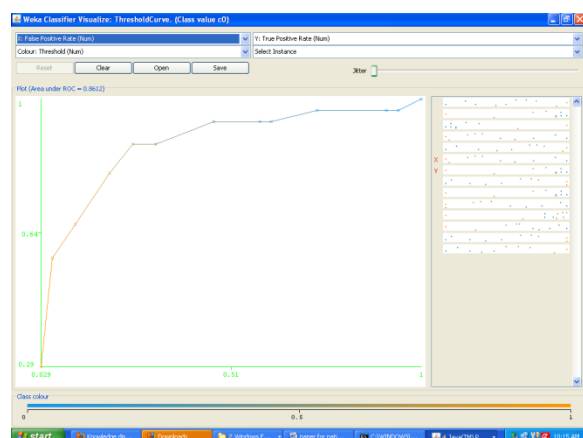


Figure 3: Visualize threshold curve for c0

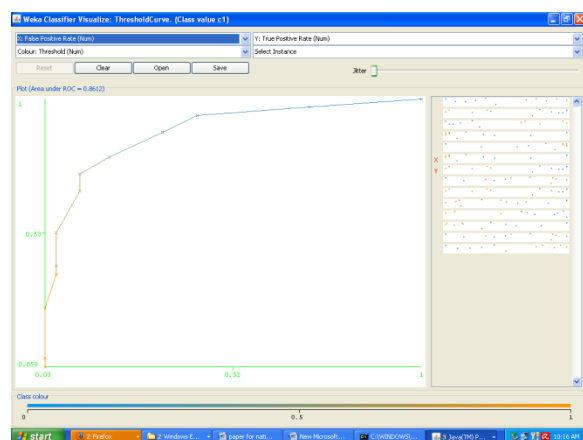


Figure 4: Visualize threshold curve for c1

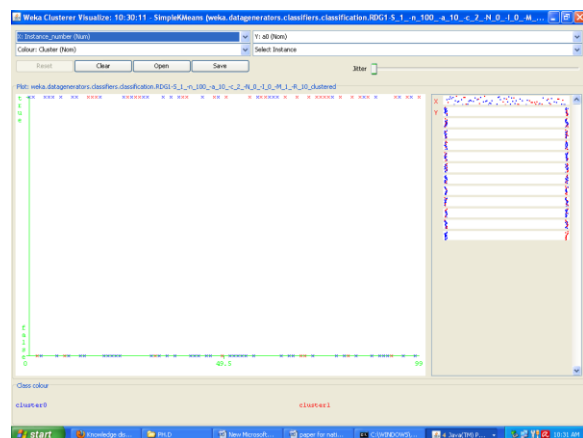


Figure 5: Visualize WEKA clusterer

First we applied Random Forest algorithm on data file poach.arff without clustering and then we combine the K-Means clustering algorithm the results are as follows:

Parameter	Random Forest	K-Means Clusterer+RF
Precision	0.807	0.84
Recall	0.81	0.86
Specificity	0.81	0.86
Sensitivity	0.255	0.24
F-Score	0.808	0.83
ErrorRate	0.265	0.126
Accuracy	81%	82.3%

Table 1: WEKA result on Random Forest Algorithm with K-Means clustering

Observations and Analysis

1. It may be observed from Table 1 that the error rate of binary classifier Random Forest with Simple KMeans Clusterer is lowest i.e. 0.126 in comparison with RandomForest classifier without clusterer i.e. 0.265, which is most desirable.

2. Accuracy of Random Forest classifier with KMeans clusterer is high i.e. 82.3% which is highly required.

3. Sensitivity (TPR) of clusters (results of integration of classification and clustering technique) is higher than that of classes. In an ideal world we want the FPR to be zero. FPR is lowest of integration of clustering and classification technique, in other words closest to the zero as compared with simple classification technique with Random Forest classifier.

4. In an ideal world we want precision value to be 1. Precision value is the proportion of true positives out of all positive results. Precision value of integration of classification and clustering technique is higher than that of simple classification with Random Forest classifier.

REFERENCES

- [1] Ng, Hwee Tou, & Chan, Yee Seng, "English Lexical Sample Task via English-Chinese Parallel Text", Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval, 2007.
- [2] Zheng-Yu Niu, Dong-Hong Ji, Chew Lim Tan, "Learning model order from labeled and unlabeled data for partially supervised classification, with application to word sense disambiguation", Computer Speech & Language, 2007.

- [3] Zhimao Lu, Ting Liu, and Sheng Li, "Combining neural networks and statistics for Chinese word sense disambiguation", In Oliver Streiter and Qin Lu, editors, ACL SIGHAN Workshop, 2004.
- [4] Leacock, C., Chodorow, M., and Miller, G. A., "Using corpus statistics and WordNet relations for sense identification", Computational Linguistics, 2000
- [5] Sin-Jae Kang, Jong-Hyeok Lee, "Ontology Word Sense Disambiguation by Using Semi-Automatically Constructed Ontology", Machine Translation Summit ,2001.
- [6] Ramakrishnan G., Bhattacharyya P. , "Word Sense Disambiguation using Semantic Nets based on WordNet" LREC, Spain, 2002
- [7] Reeve LH, Han H, Brooks AD: , "WordNet: A Lexical Database for English", Communications of the ACM, 1995.
- [8] Peng Jin, Xu Sun, Yunfang Wu, Shiwen Yu, "Word Clustering for Collocation-Based Word Sense Disambiguation", Computational Linguistics and Intelligent Text Processing, Volume: 4394, Pages: 267-274 648, 2007
- [9] A. Azzini, C. da Costa Pereira, M. Dragoni, and A. G. B. Tettamanzi, "Evolving Neural Networks for Word Sense Disambiguation", Eighth International Conference on Hybrid Intelligent Systems, 2008.
- [10] Rion Snow Sushant Prakash, Daniel Jurafsky, Andrew Y. Ng , "Learning to Merge Word Senses", Computer Science Department Stanford University, 2007.
- [11] Yoong Keok Lee and Hwee Tou Ng and Tee Kiah Chia, " Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources", Department of Computer Science National University of Singapore, 2004.
- [12] G. Miller, "Wordnet: An on-line lexical database," international Journal of Lexicography, vol. 3(4), pp. 235–244, 1990.
- [13] Y. G. I. Dhillon, J. Fan, "Efficient clustering of very large document collections in Data Mining for Scientific and Engineering Applications", R. N. R. Grossman, G. Kamath, Ed. Kluwer Academic Publishers, 2001.
- [14] J. Sedding, "Wordnet-based text document clustering," Master's thesis, University of York, 2004.
- [15] D. Yuret, "Some Experiments with a Naive Bayes WSD System", In Proceedings of the International Workshop on Evaluating Word Sense Disambiguation Systems, Senseval, 2004.
- [16] Lee YK, Ng, "An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation", Proc EMNLP ,2002
- [17] Arindam Chatterjee, Salil Joshii, Pushpak Bhattacharyya, Diptesh Kanojia and Akhlesh Meena, "A Study of the Sense Annotation Process: Man v/s Machine", International Conference on Global Wordnets ,Matsue, Japan,, Jan, 2012
- [18] Brijesh Bhat and Pushpak Bhattacharyya, "IndoWordnet and its Linking with Ontology", International Conference on Natural Language Processing (ICON 2011), Chennai, December, 2011
- [19] R. Ananthakrishnan, Jayprasad Hegde, Pushpak Bhattacharyya and M. Sasikumar, "Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation", International Joint Conference on NLP (IJCNLP08), Hyderabad, India, Jan, 2008.
- [20] Holmes, A. Donkin, I.H. Witten , " WEKA: A machine Learning work bench", In proceedings of the second Australian and New Zealand Conference on Intelligent Information Systems, 1996.
- [21] Varun Kumar, Nisha Rathee, " Knowledge discovery from database Using an integration of clustering and classification", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.3, March 2011.