International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com
Volume 2, Issue 4, April 2013 **ISSN 2319 - 4847**

Performance Enhancement of WSD Using Association Rules In WEKA

Samit Kumar[1], Dr. S. Niranjan[2]

[1]Ph.D Scholar, Computer Science & Engg.,
Mewar University, Chittorgarh,
Rajasthan, India

[2]Principal, PDM college of Engg.,
Bahadurgarh, Haryana, India

ABSTRACT

WSD methods are critical for solving natural language processing tasks like machine translation and speech processing, but also boost the performance of other tasks like text retrieval, document classification and document clustering.WSD has been described as an AI-complete problem that is, by analogy to NP-completeness in complexity theory, a problem whose difficulty is equivalent to solving central problems of artificial intelligence (AI). Its acknowledged difficulty does not originate from a single cause, but rather from a variety of factors. This paper presents a method to access the knowledge resource like WordNet. We apply association rules, a data mining approach to create our database. We create an association rules based database that will be used to deduce the sense of ambiguous word. The method uses the structured knowledge resources, which provide information about relationships between words. At last the WEKA tool used to access the association rules based database. The experiment result shows that the method has high precision and accuracy as compared to the mining without association rules.

Keywords: Association rules,Apriori, FP- Growth, Data mining, WEKA, WORDNET, BNC

1. INTRODUCTION

The problem of word sense disambiguation (WSD) is the task of automatically assigning the most appropriate meaning to a polysemous word within a given context. The problem of formalizing and quantifying the intuitive notion of similarity has a long history in philosophy, psychology, and artificial intelligence, and many different perspectives have been suggested. Recent research on the topic in computational linguistics has emphasized the perspective of semantic relatedness of two lexemes in a lexical resource, or its inverse, semantic distance. One of its applications is the word sense disambiguation. All human languages have words that can mean different things in different contexts, such words with multiple meanings are potentially “ambiguous”. For almost all applications of language technology, word sense ambiguity is a potential source of error. “Word Sense Disambiguation (WSD)” is the task of figuring out the intended meaning of a word when used in a sentence. In many ways, WSD is similar to part-of-speech tagging. It involves labeling every word in a text with a tag from a pre-specified set of tag possibilities for each word by using features of the context and other information.

“Polysemy” - a single word from having more than one meaning;

“synonymy” - multiple words having the same meaning, are both important issues in natural language processing or artificial intelligence related fields.

1.1 Wordnet

WordNet is a manually-constructed lexical system developed by George Miller at the Cognitive Science Laboratory at Princeton University. It reflects how human beings organize their lexical memories. The basic building block of WordNet is synset consisting of all the words that express a given concept[12]. WordNet is lexical database for the English language that groups English word into set of synonyms called synset. Each such Synset represents a single distinct sense or concept. For example, in Wordnet, the synset {car, auto, automobile, machine, motorcar} represents the concept of “4-wheeled motor vehicle; usually propelled by an internal combustion engine”. WordNet stores information about words belong to four parts–of–speech: nouns, verbs, adjectives and adverbs. In Wordnet 2.0, there are 146,350 words organized in 111,223 synsets, approximate 20% of the words in WordNet are polysemous; approximate 40% have one or more synonyms, some 300 prepositions, pronouns, and determiners are given no semantic illustration in WordNet. Wordnet database groups English nouns, verbs,adjectives, and adverbs into synsets that are in turn linked through semantic relations that determine word definitions and senses. WordNet 2.0 features a rich set of 333,612 relation links between synsets. Table 1 lists some semantic relations (links) in Wordnet The two most typical relations for nouns are hyponymy and hypernymy. For instance, {car, auto,automobile, machine, motorcar} are the hyponyms of {motor vehicle, automotive vehicle}, and {motor vehicle, automotive vehicle} are their hypernyms.

Table 1: Some semantic links defined in Wordnet

Semantic Relation	Meaning	Example
Hypernym	*X* is a kind of *f(X)*	Apple *is a kind of* fruit
Hyponym	*f(X)* is a kind of *X*	Zebra *is a kind of* Horse
Holonym	*X* is a part/member of *f(X)*	Wheel *is a part of* a car
Meronym	*X* has part/member *f(X)*	Table *has part* leg
Antonym	*f(X)* is the opposite of *X*	Wet *is the opposite of* dry

As each Synset in Wordnet is constructed to use terms to represent a single unique idea of the world, each synset can be viewed as a "concept". In set theory, if every member of the set A is also a member of the set B, then A is said to be a "subset" of B, written A B. In Wordnet, the hyponymy/hypernymy is a semantic relation betweenword meanings. For instance, {maple} is a hyponym of {tree}, and {tree} is a hyponym of {plant}[11]. The idea of "subset/superset" to "set" is similar to the "hyponym/hypernym" relations to "synsets" in the Wordnet. If we observe Wordnet's hypernym/hyponym relations, a "set" formed by the union of all children synsets/nodes of a father synset/node would also be a synset, which is actually the father node itself. In the same way, these children synsets/nodes are the unions of their own children synsets/nodes; and so on until to the leaf-synsets in the tree. Here the leaf-synsets are at the bottom of the Wordnet hypernym/hyponym hierarchy and are minimal "concepts" in the Wordnet hierarchy. From the opposite view point, some synsets/nodes may group together to form a synset/node that carries a more generic concept. According to above iscussions, now we investigate how to use set algebra to represent a concept/synset using some others more specific or more general concepts/synsets in the Wordnet hierarchy, yet keep the original meaning and scope unchanged[5].
The insight of our semantic relatedness measurement method is to observe the"distribution of concepts". Here, in this research, we are looking for a method (or clues) to judge the relatedness between different concepts defined in Wordnet, thus we have to specify - "in what situation are the different concepts more/less related to each others?"

1.2 WEKA
Weka[14] stands for Waikato Environment for Knowledge Analysis. It is software developed by the University of Waikato in New Zealand. Weka provides implementation of state-of -the are machine learning algorithms. It is freely available on the World Wide Web as [14]. The software is developed in Java, thus enabling porting and compatibility with various platforms. Weka provides a command line interface as well as a GUI interface. Its functionality can be divided into two main categories, Explorer and Experimenter. Using Explorer, one can preprocess a dataset, feed it into a learning scheme and analyze the resulting classifier and its performance. A learning method is called a classifier. Using Experimenter, one can apply several learners and compare their performance to choose a learner for prediction. Implementation of learning schemes is the most significant and valuable feature of Weka. Tools for preprocessing the data, called filters are also a useful feature of Weka. The main focus of Weka is on classifier and filter algorithms. It also has implementations of algorithms for learning association rules and for clustering data for which no class value is specified. Weka requires that the dataset to which the algorithms are going to be applied, should be in ARFF format. Before one can apply any algorithm to the data, the data needs to be converted into the ARFF format.

1.3 Association Rule Mining in Text Databases
The access to a large amount of textual documents becomes more and more effective due to the growth of the Web, digital libraries, technical documentation, and medical data. These textual data constitute resources that it is worth exploiting. In this way knowledge discovery [13] from textual databases, or for short, text mining (TM), is an important and difficult challenge, because of the richness and ambiguity of natural language (used in most of the available documents).Mining association rules between words in text documents has been done already in some researches. These efforts have shown that text databases cannot be efficiently analyzed by standard association mining algorithms [9]. This is because the characteristics of text databases are quite different from those of relational and transactional databases. First, the number of distinct words in a text database is usually quite large (large size of I) and so is the number of words in each document (long transactions). The large number of words implies a large number of possible patterns (sets of words) both in a document collection and in each individual document. Thus, a text AR mining algorithm needs to explore a much larger search space to find interesting patterns. Moreover, the document frequency of each word is usually very low. It was thus used in the existing work for mining text association rules. However, this will cause a large set of trivial patterns discovered. One paper aims to mine shallow word patterns on text and is fundamentally different from our task of mining relation associations on RDFs. For discovering more detailed

knowledge from text, the knowledge discovery stage, the conceptual graphs clustered into a hierarchy. Then, pattern mining techniques, such as association rule mining, can be applied on this hierarchical structure.

2. RELATED WORK

In some Wordnet based WSD approaches, semantic distance is calculated using the edge counting principle. The measurement usually used shortest path between concepts combined with other related features. The use of density is based on the observation that words in a more densely part of the hierarchy are more closely related than words in sparser areas (Agirre and Rigau, 1996). They proposed the *conceptual density* concept for WSD.

There are some hybrid approaches that combine thesaurus, with corpus statistics (Resnik, 1995; Jiang and Conrath, 1997). Resnik defines the similarity of two concepts defined in WordNet to be the maximum information *content of* their lowest super-ordinate. The information content of a concept relies on the probability of encountering an instance of the concept. (Jiang and Conrath, 1997) uses the notion of information content in the form of theconditional probability of encountering an instance of a child-synset when given an instance of a parent-synset. (Yang, C.Y. and Hung, J.C., 2005) overcome the problems of word ambiguity by indexing textual information with its underlying concepts using Wordnet and the proposed WSD method. A system designed for automatically answering student questions in an e-learning environment is designed.

(Mihalcea and Moldovan, 2000) first determine the most common sense-pairs. Subsequently, verb-noun pairs are disambiguated by taking the first t possible senses of the words and calculating "conceptual density" of the pairs by examining WordNet glosses of the sub-hierarchies. This then ranks each pair of senses by looking at noun-context of the verb and comparing it with the given noun.

(Chua, S. and Kulathuramaiyer, N., 2004) employs noun synonyms and word senses for feature selection to select terms that are semantically representative of a category of documents. The categorical sense disambiguation extends the use of WordNet.

Information retrieval using word senses is emerging as a good research challenge on semantic information retrieval. (Kim, S. B., Seo, H. C. and Rim, H. C., 2004) proposed a method using word senses in information retrieval: root sense tagging method.

3. WSD ALGORITHM BASED ON MINING ASSOCIATION RULES

Association rule mining can also be used for mining association rules from textual data with few changes. Association rules mining for textual data can use to create statistical thesaurus, to extract grammatical rules and to search large online data efficiently[10].The context of an ambiguous word is regarded as a transaction record, the words in the context and the senses of the ambiguous word are regarded as items. If some items frequently occur together in some transactions (the context of the ambiguous word), then there must be some correlation between the items. Given an ambiguous word, Y presents the set of the senses of the word, X means the set of its context words. The item sets I includes all senses of the ambiguous word and all words of its context. The database D is the context document sets of the word, each record T is a context of the word. The association rule X □□ Y means that the sense of the word can be determined by its context. The basic idea of the WSD algorithm based on mining association rules is: to discover the frequent item sets composed of the sense of the ambiguous word and its context by scanning its context database, which support degree is no less than the threshold of support degree; to produce the association rules X □□ Y which confidence degree is no less than the threshold of the confidence degree from maximum frequent item sets at last to determine the sense of the ambiguous word by choosing the sense which the most association rules deduced.

Association rules mining is a two step process: 1. is to find out the frequent itemsets which is also known as candidate items and 2. is to filter out important association rules from the candidate itemsets [2]. Identification of frequent itemsets is a resource and time consuming task and most of the research focuses that how to prune items to generate minimum valid frequent itemsets and maximum association rules. Text mining can mine association rules between letters, words, sentences and even paragraphs, they can be used for building a statistical thesaurus, and extracting phrases form text and enhancing search results. The important considerations in text mining are:

In text databases, distribution of words varies from the conventional transactional databases. Following Procedure followed to find out frequent itemsets

- Use the frequent itemsets to generate association rules.
- Find the *frequent itemsets*: the sets of items that have minimum support
  - A subset of a frequent itemset must also be a frequent itemset
    - i.e., if {*AB*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset
  - Iteratively find frequent itemsets with cardinality from 1 to *k (k*-itemset*)*
- Join Step: C_k is generated by joining $L_{k-1}$with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

Algorithm: Pseudo code

- Join Step: C_k is generated by joining $L_{k-1}$with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
- Pseudo-code:

```
Ck: Candidate itemset of size k
Lk : frequent itemset of size k
L1 = {frequent items};
for (k = 1; Lk !=∅; k++) do begin
    Ck+1 = candidates generated from Lk;
   for each transaction t in database do
      increment the count of all candidates in Ck+1 that are contained in t
   Lk+1  = candidates in Ck+1 with min_support
   end
return ∪k Lk;
```

Generation of Strong Association Rules

- For Each Frequent Item Set L, generate all non empty set of L.
- For every non empty set s of L, output rule "s=>(L-s)"
- if support_count(L) /support_count(s) >=min_conf (Strong Association rules)

e.g. subsets of set { 2,3,5} are {2},{3},{5},{2,3},2,5},{3,5}

Output Rules:

- 2=>3,5 confidence = 2/3 = 67 %
- 3=>2,5 confidence = 2/3 = 67 %
- 5=>2,3 confidence = 2/3= 67 %
- 2 ∧ 3=>5 confidence = 2/2 =100 %(Strong Rule)
- 2 ∧ 5=>3 confidence = 2/3 =67
- 3 ∧ 5=>2 confidence = 2/2 =100%(Strong Rule)

FP-Growth Method: Construction of FP-Tree

- First, create the root of the tree, labeled with "null".
- Scan the database D a second time. (First time we scanned it to create 1-itemset and then L).
- The items in each transaction are processed in L order (i.e. sorted order).
- A branch is created for each transaction with items having their support count separated by colon.
- Whenever the same node is encountered in another transaction, we just increment the support count of the common node or Prefix.
- To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links.
- Now, the problem of mining frequent patterns in database is transformed to that of mining the FP-Tree.

4. EXPERIMENTAL SETUP

Problem is to gain an understanding of the senses of the word with its relative position in the sentence. Figure 1. depicts the design of the system. The text file is pre-processed using Stanford Parser. The perl parser takes as input the file containing the pairs of context and sense. The output of the perl parser is a file in ARFF format, listing all the instances created, the features chosen and list of values, which each feature can take. The perl parser interfaces with WordNet using WordNet::QueryData, a perl interface created for WordNet. The file in ARFF format, is provided as input to the Weka software. We use the apriori algorithm and FP growth Algorithm which are the implementation of association rules. The output of the Weka software is a rule which deduce the sense of ambiguous word in respect to given context. Strong rules can be deduced by using appropriate support count and confidence level.

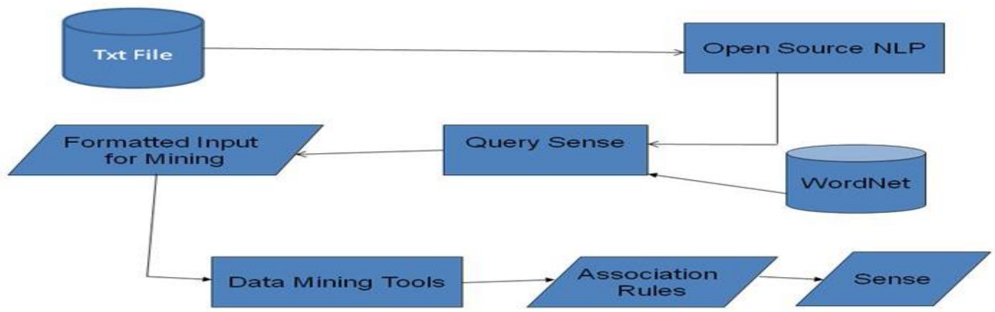

Figure 1: System Architecture

The word is considered as a whole entity and attributes and features are determined with respect to the word. Preprocessing of text is done in first step. In order to proceed with extraction, we convert the plain text data into structured data. For this, we deploy a natural language parser in our system, i.e., a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and determine which words are the subject or object of a verb. More specifically, we utilize the Stanford Parser, an open source natural processing tool written in Java. The Stanford parser reads the text file sentence by sentence and gives the result in a syntactic structure. First we need to analyze the tree output by Stanford parser. Each word in a sentence has a specific tag. Each tag is located on a node of the tree and all the other words composing the plain text data will be located in the leaves.

Figure 2: Parse Structure of a Sentence

Target word selected from the sentence. All the senses of ambiguous word are taken from WORDNET which have the word with the correct part of speech. The knowledge base for the problem is a repository from BNC [16]. Free sentences are taken from BCC. All the context words in respect to target ambiguous word are selected. Instances are created, by computing attributes and features with respect to the contexts and senses. Using these instances a transactional database for association rules is created using the context words and senses of target word. A file in ARFF format is given to algorithm in Weka. 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 bass
@attribute word-1 string
@attribute the5 nominal
@attribute meaning {music of low tone, type of fish}
@data

The output of Weka is the strong association rules. Rules specified that if the given word whenever comes in this context, the meaning of the ambiguous word can be found.

5. RESULTS

We have applied the apriori and FP growth algorithm for association rules in WEKA which is a data mining tool. Weka is a graphical user interface, through which we have implemented our algorithms. We have applied some preprocessing and filtering techniques to normalize our data.

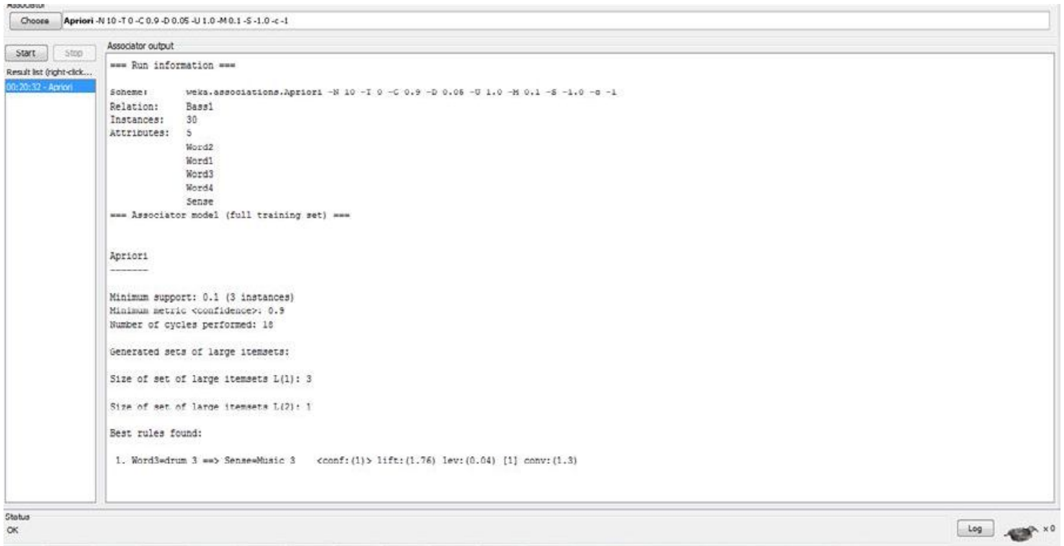

Figure 3: Weka Implementation of Association Rules

We have taken the followings performance measures to evaluate our work.

- Precision : *p* is the number of correct results divided by the number of all returned results
- Recall: *r* is the number of correct results divided by the number of results that should have been returned

The F score (also F-measure) is a measure of a test's accuracy. It considers both the precision *p* and the recall *r* of the test to compute the score :

$$F = 2.\frac{Precision.Recall}{Precision + Recall}$$

The lift of a rule is defined as $lift\, x \Rightarrow y = \frac{support\,(X \cup Y)}{support(X) \times support(Y)}$ or the ratio of the observed support to that expected if X and Y were independent.

The conviction of a rule is defined as $Conv(X \Rightarrow Y) = \frac{1 - supp(Y)}{1 - conf(X \Rightarrow Y)}$.

We applied these two algorithms first on data without sampling. Then we apply the sampling on the data file and then association rules algorithms. Following Table shows the results of our experiments.

Table 2: Results Without Stratified Data Sampling

Parameter	Apriori	FP Growth
Precision	**0.84**	**0.85**
Recall	**0.81**	**0.86**
Lift	**1.72**	**1.84**
Conv	**1.22**	**1.3**
Accuracy	**82.4%**	**85.3%**

It is observed that association rules can also applied to natural language processing problems. The results presented in above tables conclude that high accuracy can be achieved using data mining techniques. The algorithm FP growth has high performance than Apriori Algorithm. It is also noted that table 3 shows high accuracy and performance as compared to table2. By using conviction parameter, it can be predicted that above than 80% association rules shall deduce the correct sense of ambiguous word.

Table 3: Results after Sampling

Parameter	Apriori	FP Growth

Precision	**0.86**	**0.88**
Recall	**0.83**	**0.87**
Lift	**1.77**	**1.89**
Conv	**1.34**	**1.24**
Accuracy	**83%**	**86%**

6. CONCLUSION AND FUTURE SCOPE

In this paper we implemented the two algorithms of association rules for computing the meaning of an ambiguous word. We use the structured data for our experiments. It is observed that we can generate strong association rules by varying our predefined support count and confidence level. FP growth shows high performance. If we use sampling, results can be further improved. Only limited algorithms are available in WKA for association rules. In future we will apply association rules using classifiers and clustering. A fuzzy association rules can also be imposed to improve the performance of WSD system.

REFERENCES

[1] Ramakrishnan and B. Prithviraj "Soft word sense disambiguation", GWC Proceedings, PP 291-298, 2004

[2] J. Han and M. Kamber, *Data Mining Concepts and Techniques 2nd Edition*, Morgan Kaufmann Publishers, 2006

[3]Andres Montoyo, Armando Suarez, German Rigau, Manuel Palomar "Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods", Journal of Artificial Intelligence Research ,2005

[4] Chunhui Zhang, Yiming Zhou and Trevor Martin "Genetic Word Sense Disambiguation Algorithm" Second International Symposium on Intelligent Information Technology Application, IITA '08, Shanghai, 20-22 Dec. 2008

[5] Myunggwon Hwang Chang Choi Byungsu Youn Pankoo Kim "Word Sense Disambiguation Based on Relation Structure" International Conference on Advanced Language Processing and Web Information Technology, Dalian Liaoning, China ,2008,.

[6] David Martinez Iraola "Supervised Word Sense Disambiguation: Facing Current Challenges", 2004.

[7] Jiang Tao, Tan Ah-Hwee, Senior Member, IEEE, and Wang Ke, "Mining Generalized Associations of Semantic Relations from Textual Web Content", IEEE Transactions on Knowledge and Data Engineering, VOL. 19, NO. 2, FEBRUARY 2007.

[8] Gaona M.A.R., Gelbukh A., Bandyopadhyay S., "Web-Based Variant of the Lesk Approach to Word Sense Disambiguation" Eighth Mexican International Conference on Artificial Intelligence, 2009. MICAI 2009, Guanajuato, P 103 - 107.

[9] Zhang Zheng, Zhu Shu "A New Approach to Word Sense Disambiguation in MT System" World Congress on Computer Science and Information Engineering, 2009, Los Angeles, CA, P 407 – 411.

[10] M.H.Margahny and A.A.Mitwaly "Fast Algorithm for Mining Association Rules" AIML 05 Conference, 19-21 December 2005, CICC, Cairo, Egypt.

[11] Xiao Guo, Dayou Li, Clapworthy, G." A Conditional mutual information based selectional Association and Word Sense Disambiguation" International Conference on Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009, Dalian, P 1 – 7.

[12] S. G. Kolte and S. G. Bhirud "WordNet: A Knowledge Source for Word Sense Disambiguation" International Journal of Recent Trends in Engineering, Vol 2, No. 4, November 2009.

[13] Bridget T. McInnes, Ted Pedersen, Ying Liu, Serguei Pakhomov, and Genevieve B. Melto "Using Second-order Vectors in a Knowledge-based Method for Acronym Disambiguation" Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL 2011), June 23-24, 2011, pp. 145 - 153, Portland, Oregon.

[14] 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, 1994.

[15] Yang, C.Y., Hung, J.C., Wang, C.S. Chiu, M.S. and Yee G. Applying Word Sense Disambiguation to Question Answering System for E-Learning. The 19th International Conference on Advanced Information Networking and Applications (AINA 2005).

[16] http://www.natcorp.ox.ac.uk/

Authors

Samit Kumar, completed BE (CSE) in 2003, M.Tech(CSE) in 2007 and M.Phil(CS) in 2010. Pursuing PH.D. in CSE. Have total 9.5 Years of teaching experience. Presently working as Assistant Professor in CSE department in Ganga Institute of Technology and Management, Jhajjar. Attended five sponsored workshops on different emerging technologies. Having 4 publications in reputed journals and 12 papers in national and international level conferences.

S. Niranjan, did his M.Tech(Computer Engg.) from IIT Kharagpur in 1987. Completed Ph.D.(CSE) in 2004 and Ph.D.(I&CT) in 2007, Have total 26 years of teaching experience. Presently working as Principal in PDM College of Engg. Bahadurgarh(Haryana). Having 6 publications in journals and 30 other papers published in conferences.