Sentiment Analysis Methods Comparison & Comprehensive study of word2vec algorithm

[1]K. DAVID RAJU, [2]Dr. BIPIN BIHARI JAYASINGH

[1]Research Scholar, Rayalaseema University, KURNOOL

[2]Professor, Dept.of IT, CVR College of Engineering, Hyderabad

ABSTRACT-Citation sentiment analysis is an important task in scientific paper analysis. Existing machine learning techniques for citation sentiment analysis are focusing on labor-intensive feature engineering, which requires large annotated corpus. As an automatic feature extraction tool, word2vec has been successfully applied to sentiment analysis of short texts. In this work, I conducted empirical research with the question: how well does word2vec work on the sentiment analysis of citations? The proposed method constructed sentence vectors (sent2vec) by averaging the word embeddings, which were learned from Anthology Collections (ACL-Embeddings). I also investigated polarity-specific word embeddings (PS-Embeddings) for classifying positive and negative citations. The sentence vectors formed a feature space, to which the examined citation sentence was mapped to. Those features were input into classifiers (support vector machines) for supervised classification. Using 10-cross-validation scheme, evaluation was conducted on a set of annotated citations. The results showed that word embeddings are effective on classifying positive and negative citations. However, hand-crafted features performed better for the overall classification.

KEYWORDS: *Twitter, Sentiment analysis , Opinion mining, word2vec, lexicon,valence shifter,h2o*

1. INTRODUCTION

The opinions of others have a significant influence in our daily decision-making process. These decisions range from buying a product such as a smart phone to making investments to choosing a school—all decisions that affect various aspects of our daily life. Before the Internet, people would seek opinions on products and services from sources such as friends, relatives, or consumer reports. However, in the Internet era, it is much easier to collect diverse opinions from different people around the world. People look to review sites (e.g., CNET, Epinions.com), e-commerce sites (e.g., Amazon, eBay), online opinion sites (e.g., TripAdvisor, Rotten Tomatoes, Yelp) and social media (e.g., Facebook, Twitter) to get feedback on how a particular product or service may be perceived in the market. Similarly, organizations use surveys, opinion polls, and social media as a mechanism to obtain feedback on their products and services. sentiment analysis or opinion mining is the computational study of opinions, sentiments, and emotions expressed in text. The use of sentiment analysis is becoming more widely leveraged because the information it yields can result in the monetization of products and services. For example, by obtaining consumer feedback on a marketing campaign, an organization can measure the campaign's success or learn how to adjust it for greater success. Product feedback is also helpful in building better products, which can have a direct impact on revenue, as well as comparing competitor offerings.

WHAT IS SENTIMENT ANALYSIS

- Positive/Negative Polarity assigned to text
- The Sentiment 'space' is being expanded to accommodate more than a single dimension
- Classification with respect to emotion: Joy, frustration, sadness are occurring
- Classification with respect to stance (either for, or against a position) is similar to, but not entirely the same as sentiment
- Sentiment analysis is also known as opinion mining
- Sentiment Analysis is a branch of computer science, and overlaps heavily with Machine Learning, and Computational Linguistics
- Why? One seeks to understand the general opinion across many documents within a corpus (e.g., all tweets relating to a given brand).
- This is labor intensive, so we use ML to automatically label documents via classifier through a labeled dataset (supervised learning).

WHY IT IS USEFUL

- Sentiment analysis often correlates well with real world observables.
- For commercial aspects: Brand Awareness
- Stock fluctuations and public opinion
- Health related: Vaccine sentiment vs. coverage
- Public safety: Situational awareness in mass emergencies via Twitter

2. DATA COLLECTION

Data Collection Obtaining the data covers a significant part of this work. User comments in a movie site are preferred as data source. There are also ratings on the movie site, with comments from users and ratings of 1 and 5 points. The reason for choosing film critics as datasets is that there are numerical rating values given along with the user comments. Apache

ManifoldCF (MCF) was used to obtain the data. MCF technology is an application that transfers data contained in data sources to target resources. In the general use of MCF, data mapping is done by establishing a bridge to content indexing systems from enterprise content management systems (also applying content security policies during the process). In this study, MCF was used as a web browser and user comments in the designated movie site were drawn to the file system. The first step in collecting data with MCF is to set the source from which the data will be pulled. In MCF, this is called repository connector and there are many repository connectors in MCF. One of these is the web connector. The web connector passes the data of the web pages to the target sources, and if any authorization system is used before this connector is created, an appropriate idle authorization connector must be created. Using this technology, the data is drawn with the following sequence: 1. Create an empty authorization group. 2. Create a Web adapter. 3. A time adapter is created to configure which time intervals the data will be retrieved. 4. Create a file adapter to save the data. 5. After all the necessary adapters have been created, finally a transfer job needs to be created. In other words, which site to connect to and the index area (user comments) is determined. 6. Once all the work is done, the created job is started. Once all the configurations are complete, Apache ManifoldCF scans all web pages and saves the pages with user comments to the local file system in HTML format. In this way, 2983 movie reviews were collected for use as sample data sets. In order to extract the comments out of this data, the Groovy script language which can work on the JVM was used and the edited data was saved in JSON format.

3. DATA ANALYSIS

3.1 METHODS FOR SENTIMENT ANALYSIS

Lexicon Based Approach:

Lexicon based approach The lexicon based approach is based on the assumption that the contextual sentiment orientation is the sum of the sentiment orientation of each word or phrase. Turney (2002) identifies sentiments based on the semantic orientation of reviews. (Taboada et al., 2011; Melville et al., 2011; Ding et al., 2008) use lexicon based approach to extract sentiments. Sentiment Analysis on microblogs is more challenging compared to longer discourses like reviews. Major challenges for microblog sentiment analysis are short length status message, informal words, word shortening, spelling variation and emoticons. Sentiment Analysis on Twitter data have been researched by (Bifet and Frank, 2010; Bermingham and Smeaton, 2010; Pak and Paroubek, 2010). We use our lexicon based approach to extract sentiments. The open lexicon such as Sentiwordnet (Esuli and Sebastiani, 2006; Baccianella et al., 2010), Q-wordnet (Agerri and Garc´ıa-Serrano, 2010), WordNet-Affect (Strapparava and Valitutti, 2004) are developed for supporting Sentiment Analysis. Studies have been made on preprocessing tweets. Han and Baldwin (2011) used a classifier to detect word variation and match the related word. Kaufmann and Kalita (2010) gives the full preprocessing approach to convert a tweet to normal text. Sentiment Analysis on Twitter data is not confined to raw text. Analyzing Emoticons have been an interesting study. Go et al. (2009) used emoticons to classify the tweets as positive or negative and train standard classifiers such as Naive Bayes, Maximum Entropy, and Support Vector Machines. Hashtag may have some sentiment in it. Davidov et al. (2010) used 50 hashtags and 15 emoticons as sentiment labels for classification to allow diverse sentiment types for the tweet. Negation and intensifier play an important role in Sentiment Analysis. Negation word can reverse the polarity, where as intensifier increases sentiment strength. Taboada et al. (2011) studied role of the intensifier and negation in the lexicon based Sentiment Analysis. Wiegand et al. (2010) survey the role of negation in Sentiment Analysis.

3.2 Machine Learning Based Approach:

A. Machine Learning Algorithms – The machine learning algorithm is a branch of artificial intelligence. It focuses on building models that have the ability to learn from data. “Machine Learning is a field of study, which gives computers the ability to learn without being explicitly programmed”. A supervised learning algorithm learns to map the input examples to expected target. The machine learning algorithm should be able to generalize the training data after the correct implementation of the training process, So that it can accurately map new data that it has never seen before. Naïve Bayes The Naïve Bayes classifier is a simple probabilistic model which relies on the assumption of feature independent in order to classify input data. The algorithm is commonly used for text classification. It has simpler implementation, low computational cost and its relatively high accuracy. The algorithm will take every word in the training set and calculate the probability of it being in each class (positive or negative). Then the algorithm is ready to classify new data. When a new sentence is being classified it will split it into single word features. The model will use the probabilities, which computed in the training phase to calculate the condition probabilities of the combined features in order to predict its class. The advantage of the Naïve Bayes classifier is that it utilizes all the evidence that is available to it in order to make a classification. Using this approach it takes into account that many weak features which may have relativistic minor effect individuals may have a much larger influence on the overall classification when combined .

Support Vector Machine:

The support vector machine is considered as a non-probabilistic binary linear classifier. It works by plotting the training data in multidimensional space; it then tries to separate the classes with a hyperplane. If the classes are not immediately linearly separable in the multidimensional space the algorithm will add a new dimension in an attempt to further separate the classes. The SVM algorithm chooses the hyperplane which provides the maximum separation between the classes has the greatest margin or the maximal margin hyperplane which minimizes the upper bound of the classification errors. A standard method for finding the optimum way of separating the classes is to plot two hyperplanes in a way that there are no data points between them, and then by using these planes the final hyperplane can be calculated. The data points that fall on these planes are known as the supports. A major problem with the SVM is that by adding extra dimensions the size of the feature space increases. From a processing point of view the SVM algorithm counteracts this by using dot products in the original space. This method hugely reduces processing as all the calculations are performed in the original space and then mapped to the feature space.

Decision Tree:

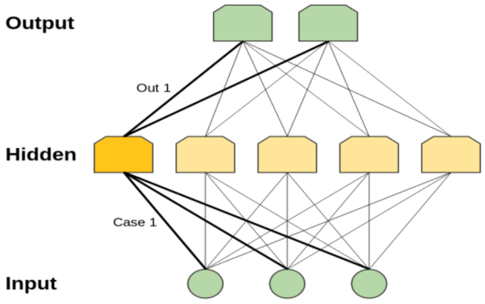
Decision trees on one of the most widely used machine learning algorithms which can be adapted to almost any type of data. It divides its training data into smaller parts in order to identify patterns that can be used for classification. Then, the knowledge is represented in the form of logical structure similar to a flow chart that can be easily understood without any statistical knowledge. The algorithm is particularly used where many hierarchical categorical distinctions can be made. The structure of a decision tree consists of a root node which represents the entire dataset, decision nodes, which perform the computation and leaf nodes which produce the classification. In the training phase the algorithm learns what decisions have to be made in order to split the labeled training data into its classes . By passing data through the tree, unknown instance is classified. At each decision node a specific feature of the input data is compared with a constant that was identified in the training phase. The computation which takes place in each decision node usually compares the selected feature with this predetermined constant, the decision will be based on whether the feature is greater than or less than the constant, creating a two way split in the tree. The data will eventually pass through these decision nodes until it reaches a leaf node which represents its assigned class.

3.3 Deep learning based Approach

A neural network with a hidden layer has universality: given enough hidden units, it can approximate any function. This is a frequently quoted – and even more frequently, misunderstood and applied – theorem.

It's true, essentially, because the hidden layer can be used as a lookup table.

For simplicity, let's consider a perception network. A perception is a very simple neuron that fires if it exceeds a certain threshold and doesn't fire if it doesn't reach that threshold. A perception network gets binary (0 and 1) inputs and gives binary outputs.



Note that there are only a finite number of possible inputs. For each possible input, we can construct a neuron in the hidden layer that fires for that input [1], and only on that specific input. Then we can use the connections between that neuron and the output neurons to control the output in that specific case.

And so, it's true that one hidden layer neural networks are universal. But there isn't anything particularly impressive or exciting about that. Saying that your model can do the same thing as a lookup table isn't a very strong argument for it. It just means it isn't *impossible* for your model to do the task. Universality means that a network can fit to any training data you give it. It doesn't mean that it will interpolate to new data points in a reasonable way. No, universality isn't an explanation for why neural networks work so well. The real reason seems to be something much more subtle… And, to understand it, we'll first need to understand some concrete results.

4 LEVELS OF SENTIMENT ANALYSIS

Three different levels on which sentiment analysis can be performed depending upon the granularities required are:

A. DOCUMENT LEVEL

The document level sentiment analysis classifies the entire document opinion into different sentiment, for aproduct or service. This level classifies opinion document into a positive, negative or neutral sentiment.

This is the simplest form of classification. The whole document of opinionated text is considered as basic unit of information. It is assumed that document is having opinion about single object only (film, book or hotel). This approach is not suitable if document contains opinions about different objects as in forums and blogs. Classification for full document is done as positive or negative. Irrelevant sentences need to be eliminated before processing.

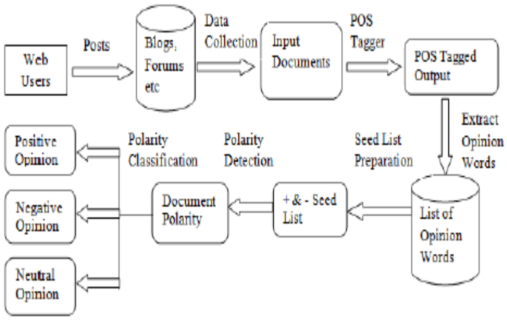


Fig. 1 Document based Sentiment Orientation System

B. SENTENCE LEVEL

The sentence level sentiment analysis determines whether each sentence expresses a positive, negative or neutralopinion, for a product or service. This type is used for reviews and comments that contain one sentence and written by the user.

C. SENTIMENT AT WORD LEVEL

In this section we present the new method for word-level sentiment analysis. We start, in section 3.1, by presenting the underlying philosophy of the method and then, in section 3.2, we present the sequential classification model and the way in which it is trained. Finally, in section 3.3, we explain how the sequence of word labels can be used to identify the sentiment expressed for specific entities.

4.1 WORD2VEC METHOD

Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Word2vec was created by a team of researchers led by Tomas Mikolov at Google. The algorithm has been subsequently analyzed and explained by other researchers.Embedding vectors created using the Word2vec algorithm have many advantages compared to earlier algorithms such as latent semantic analysis.

CBOW and skip grams

Word2vec can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram. In the continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction (bag-of-words assumption). In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words.[1][4] According to the authors' note,[5] CBOW is faster while skip-gram is slower but does a better job for infrequent words.

Parameterization

Results of word2vec training can be sensitive to parameterization. The following are some important parameters in word2vec training.

Training algorithm

A Word2vec model can be trained with hierarchical softmax and/or negative sampling. To approximate the conditional log-likelihood a model seeks to maximize, the hierarchical softmax method uses a Huffman tree to reduce calculation. The negative sampling method, on the other hand, approaches the maximization problem by minimizing the log-likelihood of sampled negative instances. According to the authors, hierarchical softmax works better for infrequent words while negative sampling works better for frequent words and better with low dimensional vectors.As training epochs increase, hierarchical softmax stops being useful.

Sub-sampling

High frequency words often provide little information. Words with frequency above a certain threshold may be subsampled to increase training speed.

Dimensionality

Quality of word embedding increases with higher dimensionality. But after reaching some point, marginal gain will diminish. Typically, the dimensionality of the vectors is set to be between 100 and 1,000.

Context window

The size of the context window determines how many words before and after a given word would be included as context words of the given word. According to the authors' note, the recommended value is 10 for skip-gram and 5 for CBOW.

Extensions

An extension of word2vec to construct embeddings from entire documents (rather than the individual words) has been proposed.[7] This extension is called paragraph2vec or doc2vec and has been implemented in the C, Python[2][6] and Java/Scala tools (see below), with the Java and Python versions also supporting inference of document embeddings on new, unseen documents.

Word vectors for bionformatic: Bio Vectors

An extension of word vectors for n-grams in biological sequences (e.g. DNA, RNA, and Proteins) for bioinformatics applications have been proposed by Asgari and Mofrad.Named bio-vectors (BioVec) to refer to biological sequences in general with protein-vectors (ProtVec) for proteins (amino-acid sequences) and gene-vectors (GeneVec) for gene sequences, this representation can be widely used in applications of machine learning in proteomics and genomics. The results suggest that BioVectors can characterize biological sequences in terms of biochemical and biophysical interpretations of the underlying patterns.

Word vectors for Radiology: Intelligent Word Embedding (IWE):

An extension of word vectors for creating a dense vector representation of unstructured radiology reports has been proposed by Banerjee et. al.[1, 3] One of the biggest challenges with Word2Vec is how to handle unknown or out-of-vocabulary (OOV) words and morphologically similar words. This can particularly be an issue in domains like medicine where synonyms and related words can be used depending on the preferred style of radiologist, and words may have been used infrequently in a large corpus. If the word2vec model has not encountered a particular word before, it will be forced to use a random vector, which is generally far from its ideal representation.

IWE combines Word2vec with a semantic dictionary mapping technique to tackle the major challenges of information extraction from clinical texts, which include ambiguity of free text narrative style, lexical variations, use of ungrammatical and telegraphic phases, arbitrary ordering of words, and frequent appearance of abbreviations and acronyms. Of particular interest, the IWE model (trained on the one institutional dataset) successfully translated to a different institutional dataset which demonstrates good generalizability of the approach across institutions.

Analysis:

The reasons for successful word embedding learning in the word2vec framework are poorly understood. Goldberg and Levy point out that the word2vec objective function causes words that occur in similar contexts to have similar embeddings (as measured by cosine similarity) and note that this is in line with J. R. Firth's distributional hypothesis. However, they note that this explanation is "very hand-wavy" and argue that a more formal explanation would be preferable.

Levy et al. (2015) show that much of the superior performance of word2vec or similar embeddings in downstream tasks is not a result of the models per se, but of the choice of specific hyperparameters. Transferring these hyperparameters to more 'traditional' approaches yields similar performances in downstream tasks.

5. EVALUATION OF SENTIMENT CLASSIFICATION ACCURACY:

5.1 Methodology

Pre-processing

The Sentence Model provided by Ling Pipe was used to segment raw text into its constituent sentences 3.The data I used to train the vectors has noise. For example, there are incomplete sentences mistakenly detected (e.g. Publication Year.). To address this issue, I eliminated sentences with less than three words.

5.2 Overall Sent2vec Training

In the work, I constructed sentence embeddings based on word embed-dings. I simply averaged the vectors of the words in one sentence to obtain sentence embeddings (sent2vec). The main process in this step is to learn the word embedding matrix Ww:

$$V_{sent2vec}(w) = \frac{1}{n} \sum W_w^{x_i} \;\; (1)$$

5.3 Polarity-Specific Word Representation Training

To improve sentiment citation classification results, I trained polarity specific word embeddings (PS-Embeddings), which were inspired by the Sentiment-SpecificWord Embedding. After obtaining the PS-Embeddings, I used the same scheme to average the vectors in one sentence according to the sent2vec model.

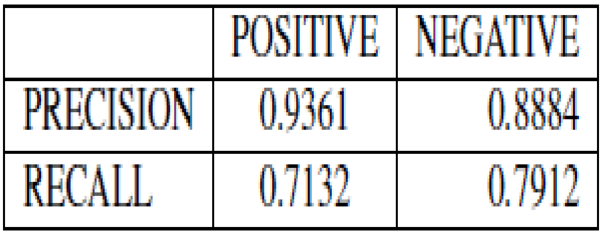
Confusion metrics:

Accuracy = (TP+TN)/(TP+TN+FP+FN)

6. RESULTS AND DISCUSSION

The result shows that word2vec approach is more accurate for larger datasets in comparison with Lexicon methods.Our sentiment engine performed reasonably well.Please see Table 1 for Precision and Recall measurements.The recall rates are lower because of our lexiconslack of coverage of all the sentiment words. Informallanguage of tweets posed another challengefor identifying negative sentiments. For example,swear words such as “sh*t” and “f**k” are generallyconsidered as negative sentiment words. Phrasessuch as “This sh*t is good” and “F**king awesome”were identified as negative sentiments when in factthey were expressing positive sentiments.

Table 1: Results



	POSITIVE	NEGATIVE
PRECISION	0.9361	0.8884
RECALL	0.7132	0.7912

The Serendio lexicon that we used has sentimentwords with a sentiment attached to it. By integratingwith a lexical source such as Sentiwordnet, wefeel we could get a more nuanced word sense disambiguation.For example, the word “good” is consideredto have positive polarity. According to Sentiwordnet3.0, good as an adjective has 21 differentsenses with different sentiments. For example, thesentiment word “good” in the phrase “A good milefrom here” gives an objective sense, not in a positivesense.

7. CONCLUSION

In this paper we presented our system that we usedfor the SemEval-2013 Task 2 for doing SentimentAnalysis for Twitter data. We got an F-score of0.8004 on the test data set.We presented a lexicon based method for SentimentAnalysis with Twitter data. We provided practicalapproaches to identifying and extracting sentimentsfrom emoticons and hashtags. We also provideda method to convert non-grammatical words togrammatical words and normalize non-root to rootwords to extract sentiments.A lexicon based approach is a simple, viable andpractical approach to Sentiment Analysis of Twitterdata without a need for training. A Lexicon basedapproach is as good as the lexicon it uses. To achievebetter results, word sense disambiguation should becombined with the existing lexicon approach.

REFERENCES

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichletalloca-tion. the Journal of machine Learning research, 3:993{1022, 2003.

[2] Jorge Carrillo de Albornoz, Laura Plaza, Pablo Gerv_as, and Alberto D__az.A joint model of feature mining and sentiment analysis for product reviewrating. In Advances in Information Retrieval, pages 55{66. Springer, 2011.

[3] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. InProceedings of the National Conference on Arti_cial Intelligence, pages 755{760. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press;1999, 2004.

[4] Thomas Hofmann. Probabilistic latent semantic indexing. In Proceedingsof the 22nd annual international ACM SIGIR conference on Research anddevelopment in information retrieval, pages 50{57. ACM, 1999.

[5] Aurangzeb Khan, BaharumBaharudin, and Khairullah Khan. Sentimentclassi_cation from online customer reviews using lexical contextual sentencestructure. In Software Engineering and Computer Systems, pages 317{331.Springer, 2011.

[6] SamanehMoghaddam and Martin Ester. Opinion digger: an unsupervisedopinion miner from unstructured product reviews. In Proceedings of the 19thACM international conference on Information and knowledge management,pages 1825{1828. ACM, 2010.

[7] SamanehMoghaddam and Martin Ester. Ilda: interdependent lda modelfor learning latent aspects and their ratings from online product reviews. InProceedings of the 34th international ACM SIGIR conference on Researchand development in Information Retrieval, pages 665{674. ACM, 2011.

KOLLURI DAVID RAJU is a Ph.D Research Scholar at the Rayalaseema University, Kurnool, A.P, INDIA. He Received M.Tech degree in COMPUTER SCIENCE & ENGINEERING in the year 2010 and B.Tech degree in COMPUTER SCIENCE & ENGINEERING in the year 2002 from JNTUH Hyderabad, TS, INDIA and He Received Diploma in COMPUTER SCIENCE & ENGINEERING in the year 1996 from VMR PolyTechnic, Rampur, SBTET, TS,INDIA. He is currently working As ASSOCIATE PROFESSOR, Department of COMPUTER SCIENCE & ENGINEERING at St.peters Engineering college, Hyderabad, TS, INDIA. He is having 15 years of Teaching experience and 2 years of Industrial experience, published more than 10 papers in National/International Journals/Conferences. He is a Member of IEEE and IAENG(International association of Engineers), ISTE(Indian Society For Technical Education), . His areas of research include Data Mining & Data Warehousing, Big Data, Machine Learning, Data Structures & Algorithms and Programming Languages.

Dr. BIPIN BIHARI JAYASINGH is Working as Professor at CVR College of Engineering , Hyderabad, TS. He Received M.Tech Degree from JNTUH and Ph.D Degree from Berhampur University.Published 51 research papers in various national/international conferences and journals. Junior Research Fellow (JRF) of Directorate of Forensic Science, Govt. of India. Nominee for young scientist award in the Indian Science Congress Association (ISCA 2006), held in Acharya N.G. Ranga Agriculture University, Hyderabad, Jan. 3 – 7, 2006. Completed Ph. D(comp. Sc.) at the age of 31 & Accepted Professor position at CVR College of Engineering, Ibrahimpatan(M), Hyderabad, RR Dist-501510, India. Guiding 7 Ph.D students. Editor, International Journal of Computer Applications in Engineering, Technology and Sciences (IJ_CA_ETS), India, ISSN: 0974-3596 since 2008. Program Committee Member, International Joint Journal Conference in Engineering 2009 (IJJCE-2009). Program Committee Member, International conference on Advances in Recent Technologies in Communication&Computing(ARTCom-2009),October-2009