Linguistic Knowledge Based Supervised Key-phrase Extraction

Tanmoy Pal[1], Haider Banka[2],Pabitra Mitra[3], Barnan Das[4]
[1,2]Department of Computer Science and Engineering, Indian School of Mines, Dhanbad, India.
[3]Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India.
[4]School of Electrical Engineering and Computer Science, Washington State University, Pullman, USA
[1]tanmoypal@aol.in,[2]banka.h.cse.@ismdhanbad.ac.in,[3]pabitra@iitkgp.ac.in,[4]barnandas@wsu.edu

ABSTRACT
The most important information about the content of a document is represented by the key phrases of that document. In this study an automatic key phrase extraction algorithm is devised using machine learning technique. The proposed method not only considers the document level statistics like TFxIDF, the linguistic features of the phrases are also incorporated. Experiment has been performed on Naïve Bayes' classifier, J48 Decision Tree, and IBk lazy learner to choose the most suitable learning model. The imbalanced class distribution problem is resolved by over sampling on the minority class samples synthetically. The experimental result reveals the accuracy and efficiency of the proposed technique.

Keywords

Key phrase extraction, machine learning, imbalance learning, sampling.

1. INTRODUCTION

1.1 Importance of the key phrase extraction

With the increasing popularity of digital media, searching over the internet to gather information has become a common task. But for efficient extraction of information many other related and relevant topics should also be looked into. A very common example is Wikipedia pages, where, along with the information of a particular topic, there are also some keywords and key-phrases which have forwarded links that gives a better understanding of the word or the phrase. Appropriate key-phrases can serve as a highly condensed summary for a document. Different academic journals & conferences ask authors to provide a list of key words or key- phrases of their work that would help in proper indexing of those documents over the web and thus can reach the person, who queries for related work, in a more precise manner. Although there are some domain-specific controlled vocabularies for almost every field of study, indexing different key-phrase for the same document is most likely to be different among individuals as this task is partially accomplished by an individual's cognition. Again, it is a very tedious job for the authors. As a solution to this problem, we automate the process of key-phrase - extraction. Key-phrase extraction has a higher degree of usage in the field of Computer Science especially in Information Retrieval & Natural Language Processing. It can be used in the field of text summarization, author assistance, index generation, query refinement, etc.

1.2 Related Work

Quite a number of research groups have been working on automated key-phrase extraction, although devising their unique solution to the problem. Different heuristics are used by [1] to extract key-phrases from a document based on syntactic clues, such as the use of italics, the presence of phrases in section headers, and the use of acronyms. But, this approach produces long list of key-phrases having very low precision. An unsupervised learning technique based on Adaptive Resonance Theory (ART) of neural networks to discover two-word key-phrases can also be used as defined in [2]. Thus, this is not applicable to one word or multiple word key-phrases**.** A supervised-learning approach combined with genetic algorithm used in [3]. In the supervised learning approach, all the candidate words in the training set are classified and tagged as keyword or non keyword by an oracle or teacher. This training set is fed to a learning model which creates its own rules and hypotheses for classification. Next, the learned model is used on new documents to tag the keywords by classifying every word individually. But it does not consider the linguistic features of the language. A Naïve Bayes' learning method has been used in [4] to get better result. Although linguistic knowledge is incorporated in [5], the relationship between the POS tags are not taken into consideration and the experiment does not suggest any specific optimal classifier to classify the phrases. Graph Based approach is the most widely used unsupervised approach. TextRank is one of the approaches proposed by [6]. As a first step, TextRank converts the document to a graph. This approach is based on the concept that an important word or sentence is related to other important sentences. When one vertex links to another one, it is basically casting a vote for the other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. But the problem with this approach is that for a large set of document it is very costly to construct the graph.

A method to extract candidates has been used in [7], which uses a simple dictionary lookup to assign basic POS-tags and a shallow parser to identify all nouns with zero or more pre-modifying adjectives and nouns. But it suffers from the shortcomings of any other dictionary look up method i.e. unavailability of dictionary in all the fields of a subject. A new approach has been introduced in [8], where to get higher conflation rate each extracted phrase is converted to a pseudo phrase. In the final step, they apply a pattern based technique to establish semantic roles and relations among remaining phrases.

A machine learning based system was proposed by [9] for the same task. They focused attention not on term conflation but on distinguishing terms specific for a given field from generic ones. For this purpose they use a measure called dispersion computed for each candidate noun phrase. The lower the dispersion, the higher the probability that this term is content bearing & possible key-phrase. A generic algorithm is used to determine the best cutoff value of dispersion. But the performance of this approach highly depends on the objective function of the genetic algorithm.

1.3 Advantages of the proposed approach

In our approach the phrase extraction problem is treated as a classification problem, where each phrase of a document will be classified either as a key phrase or non key phrase. We use different supervised machine learning technique for this purpose. While other supervised models only consider document level statistics of the document to construct the models, we also consider the linguistic feature. By analyzing this linguistic knowledge we form our model with a combination of statistical and linguistic features. To find out the best suitable classifier for this problem we did a comparative analysis between three different classifiers. As most of the phrases of a document are non key phrase, so learning model will get enough instances for negative class i.e. non key phrase, but very small amount of positive class instances i.e. key phrase. This problem in machine learning is called Imbalance Learning which is also taken care by using sampling.

The rest of the paper is organized as follows: Section 2 shows the architecture of the system, the details of the features used for learning and a short description of the different learning models used. Section 3 describes the dataset and the performance measurement parameters. Section 4 describes the experimental results and analysis along with the sampling method to solve imbalance learning. Finally Section 5 concludes the article.

2. PROPOSED APPROACH

2.1 System architecture

Fig 1 shows the architecture of our system. The input to the system is a document collection of 2000 abstracts in English, with their corresponding keywords from the *Inspec* database. This document collection is divided into two parts used separately for training and testing. We perform over the input document collection. The preprocessing stage is further described in detail in Fig 2. During preprocessing we performed input text cleaning, phrasing or tokenization. Next the stop words are removed from the documents. We use a Hybrid Stemmer to stem the phrases. Hybrid Stemmer is a combination of Porter's Stemmer and Lovin's Stemmer along

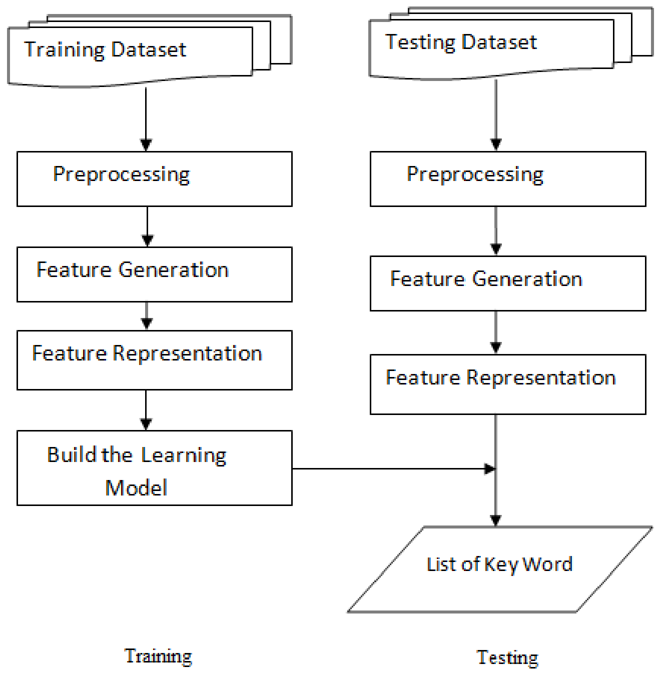

Figure 1: System Architecture.

with a modified version of stemming to get back the original word. Then we performed Parts Of Speech (POS) tagging using Conditional Random Field (CRF) Tagger. On the basis of this linguistic feature we removed some phrases which are less likely to be a key phrase. Now we generate the feature value for the features like TF x IDF, Relative Position, POS tag of the phrase and represent those feature value in a suitable format for the learning model. Then we build the machine learning model over this feature value by using the training dataset. During testing using the same procedure of training we represent the testing documents as a set of feature value and using the learned system we can extract the list of key phrases. The details of the preprocessing stages are shown in Fig 2.

2.2 System Description

Text Cleaning -The ASCII Input files are split into tokens and some other cleaning operations are also applied (a) Punctuation marks, brackets, numbers, special characters are replaced by phrase boundary.(b) Apostrophes are removed (c) Hyphenated words are split into two, (d) Single letter words are removed, (e) remaining non-token characters are deleted.

Phrasing - A key word is defined as *"Word which succinctly and accurately describes the subject or the aspect of the subject discussed in the document*" in **[10]** & "*Key phrases provide semantic metadata that summarize and characterize documents*" in **[8].** Whereas a Key Phrase is a multi word lexemes, like Machine Learning is a key Phrase having two keyword Machine and Learning. A key phrase with a single word is nothing but a key word. Key Phrases longer than one word are potentially more useful for key-phrase search and

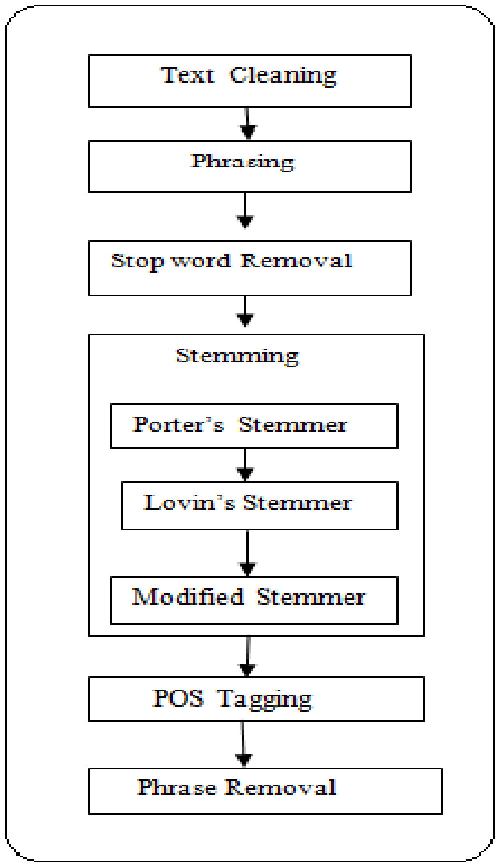

Figure 2: Preprocessing Steps.

extraction. Analysis of query logs of search engine shows that people tend to use longer phrases in order to describe the subject they are looking for, (the average query length in an AltaVista query log as 2.4 terms as estimated by [11]).That's why we are working with key phrase though some journal use the term keyword or key word. So form the bag of words or tokens we need to form the phrases. Here we used a very simple approach. We formed a list of all possible Unigram, Bigram and Trigram from the input dataset . For example the sentence " *Software cost estimation is still an open challenge*" will have the following Unigrams, Bigrams, Trigrams *Software, software cost, software cost estimation, cost, cost estimation, cost estimation is, estimation, estimation is, estimation is still, is, is still, is still open, still, still an, still an open , an, an open, an open challenge.*As most of the manually indexed key phrase is having less than or equal to three word as described in [11], that is why we only consider up to Trigram. So after this stage we got a set of phrases from the ASCII file input.

Case Folding - All the phrases are converted to either lower or upper case for easy and efficient comparison between the words or phrases.

Stop Word Removal - Stop words are the non information bearing words of a document. As there is no such fixed stop word list, we used a manual list of stop word containing 319 stop words. But the challenge in this stage was how to handle the phrases? For a unigram we just performed exact string matching with the stop word list and the phrase itself. If it matched then we removed the phrase. For bigram if either of the two words matched with any stop word in the stop word list, we remove the entire phrase. For trigram if either the beginning or the ending word matched with any of the stop word in the list, then the trigram is removed. So in the previous example the following phrases are removed - *is, an , estimation is ,is still, still an, an open, cost estimation is, is still open, an open challenge.* Here almost half of the phrases are removed. Phrases which are having stop word as middle word is correctly not removed like *estimation is still or still an open.*

Stemming - One of the main problems in automatic text processing is term variation. While humans easily identify morphologically related words as the same concept, an algorithm needs a pre-processing operation that conflates these words to the same surface form. Common variations are inflectional affixes, alternative spellings, spelling errors, and abbreviation forms. To solve this problem we used stemming. Stemming is a language processing task to reduce the inflectional words to their root or stem word. For example a human can easily identify that the word science, scientific, scientist are related to science. But how to make the machine about this? We used stemming for this purpose. In this experiment we used a hybrid stemmer, which is a combination of Porter's Stemmer, Lovin's stemmer and a modified program to get back the original words form the stemmed word. These three operations are performed sequentially. Stepwise stemming phrases on a set of words are listed in the Table I.

Original Word	Porter's Stemmed word	Lovin's Stemmed word	Final word
jealousness jealousy	jealous jealousi	Jeal jealous	jealousy jealousy
realistic reality	realist reality	Re realit	reality reality
psychology psychologists	psychology psychologists	Psycholog psycholog	psychology psychology
science scientist scientific	scienc scientist scientif	Scienc sci scientif	science science science

Table I: Stemming Stages applied to some words.

Phrase Removal - Here we performed a linguistic technique to reduce the no of phrases for the next stages. We did POS tagging of all the phrase using CRF Tagger [12] to get the POS tags of all the phrases. Now according to linguistic knowledge the noun phrases i.e a phrase whose head is a noun or a pronoun, are most likely to be a key phrase. So we found out all the noun phrases. The noun phrases which do not satisfy the regular expression of the POS tag of a phrase as described in [13], are removed. The regular expression is in the form of

(NN|NNS|NNP|NNPS|JJ)*(NN|NNS|NNP|NNPS|VBG)

This pattern means zero or more nouns or adjectives, followed by one final noun or gerund. More precisely, NN represents a singular noun, NNS represents a plural noun, NNP represents a singular proper noun, NNPS represents a plural proper noun, JJ represents an adjective, and VBG represents a gerund.

2.3 Features of the Key Phrase Extraction

Features are the individual measurable heuristic properties of the phenomena being observed. Choosing discriminating and independent features is the key to any pattern recognition algorithm being successful in classification. We used three features, two document level statistical feature-TF X IDF, Relative Position and one linguistic feature- POS tag.

TF x IDF - Term Frequency (TF) of a phrase measures the number of occurrences of a phrase in a document. Term Frequency suffers from a critical problem: all phrase are considered equally important when it comes to assessing relevancy on a query. For attenuating the effect of phrase that occur too often in the collection to be meaningful for relevance determination, we scale down the phrase weights of the phrase with high collection frequency, defined to be the total number of occurrences of a phrase in the collection. To discriminate the documents for the purpose of scoring it is better to use document level statistics than to use collection wide statistics of a phrase as described in [14].The document frequency of a phrase is defined to be the number of documents in the collection that contain that phrase. The TF of a phrase p in a document d is –

$$TF_{(p,d)} = \frac{freq(p,d)}{size(d)}$$

where freq (p,d) is the Term Frequency of the phrase p in document d and size(d) is the number of phrases in the document d.The IDF of a phrase in a document d is -

$$IDF_p = -\log_2(\frac{N}{DF_p})$$

where N is the number of phrases in the total collection, DF_p is the number of document in the collection where the term p presents. Logarithm is taken to normalize the value.

The idf of a rare term is high, whereas the idf of a frequent term is likely to be low. The tf-idf weighting scheme assigns to phrase p a weight in document d given by-

$$TF - IDF = TF_{(p,d)} * IDF_p$$

The value of this TF X IDF is varied according to the following rule –

1. Highest when p occurs many times within a small number of documents.
2. Lower when the term occurs fewer times in a document, or occurs in many documents
3. Lowest when the term occurs in virtually all documents.

Relative Position :- As most of the key phrases occur at the beginning of the document or at the last. So the position of a phrase in a document is very important. Relative position is calculated as the number of words that precede the phrase's first appearance, divided by the number of words in the document.

$$RP_{(p,d)} = \frac{m}{size(d)}$$

where size(d) is the number of phrases in the document d and m is the number of phrases p precedes in the document d.

POS Tag :- As we wanted to incorporate linguistic feature of a phrase so we choose POS tag, which is one of the important feature. We used CRF Tagger[12] for tagging the phrases. Though there are only 8-9 basic POS tags, but these tags are fine tuned to get more accurate parts of speech. According to Pen Tree Bank Project there are 36 different POS tags. For example Noun(NN) can be further classified as plural noun(NNS),proper noun singular(NNP),proper noun plural(NNPS). As we considered phrases, so for each phrase we got a set of POS tag for example - software /NN cost /NN estimation /NN .

2.4 Learning Models

Naïve Bayes' Classifier : A Bayes' classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A Naive Bayes' classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Naive Bayes' makes the assumption that the feature values are independent.

Instance Based K Nearest Neighbor : Instanced based K Nearest Neighbor works on the basic principle that during training we only places the training sample in the feature space along with their class. During testing when a new sample comes we place it in the feature space. Find the k nearest neighbors of that sample, then ask the k neighbors to vote for the new sample. Depending on the number of the votes we tag the new sample to the highest voted class.

J48 Decision Tree : J48 is an open source Java implementation of the C4.5 algorithm. A tree where each branching node represents a choice between two or more alternatives, with every branching node being part of a path to a leaf node. The leaf node represents a particular classification of a data instance, based on the given set of attributes that define the instance of data. The decision tree is initially constructed from a set of pre-classified data or training data.

3. DATA SET AND PERFORMANCE MEASUREMENT

3.1 The Corpus

The Dataset we used for the experiments in this paper consists of 2000 abstracts in English, with their corresponding keywords from the *Inspec* database[#]. The abstracts are from the years 1998 to 2002, from journal papers, and from the disciplines *Computers and Control*, and *Information Technology*. Each abstract has two sets of keywords—assigned by a professional indexer associated to them: a set of controlled terms, i.e., terms restricted to the Inspec thesaurus(in the form of .CONTR file format); and a set of uncontrolled terms that can be any suitable terms. Both the controlled terms and the uncontrolled terms may or may not be present in the abstracts (in the form of .UNCONTR file format). However, the indexers had access to the full length documents when assigning the keywords. For the experiments described here, only the uncontrolled terms were considered, as these to a larger extent are present in the abstracts (76.2% as opposed to 18.1%). The set of abstracts was arbitrarily divided into two sets: a training set (to construct the model) consisting of 1000 documents, and a test set (to get unbiased results) with the remaining 1000 abstracts. The set of manually assigned keywords were then removed from the documents.

3.2 Performance Measures

In our experiment to measure the performance of the learning system, we found out whether a phrase is correctly classified as a key phrase or a non key phrase. It might happen that a key phrase is detected as non key phrase and vice versa. So we used Confusion matrix as performance metric to reflect all this issue correctly. A confusion matrix contains information about actual and predicted classifications done by a classification system as shown in Table II.

Table II : Confusion Matrix

	Actual Outcome (Classified by Human)	
Predicted Outcome (Classified by the Machine)	True Positive(a)	False Positive(b)
	False Negative(c)	True Negative(d)

True Positive(TP):- Number of correct predictions that an instance is positive.
False Positive(FP):- Number of incorrect predictions that an instance is positive.
True Negative(TN):-Number of correct predictions that an instance is negative.
False Negative(FN):-number of incorrect predictions that an instance negative.

[#]The dataset is available at http://github.com/snkim/AutomaticKeyphraseExtraction

The True Positive (TPR) Rate here represents the percentage of phrases that are correctly classified as key phrase; the True Negative (TNR) Rate here represents the percentage of phrases that are correctly classified as non key phrase. Accuracy (AC) can be measured as the percentage of the correctly classified key and non key phrases with respect to the total no. of phrases. Recall measured what fraction of the key phrases are correctly classified as key phrase. Precision measured what fraction of phrases classified as the key phrase is really key phrase There is a well-known trade-off between precision and recall. We can optimize one at the expense of the other. We want a performance measure that yields a high score only when precision and recall are balanced. A measure that is widely used in information retrieval is the F-measure as described in [12].

$$\mathrm{F-measure} = \frac{2 * \mathrm{Precission} * \mathrm{Recall}}{\mathrm{Precission} + \mathrm{Recall}}$$

A method to evaluate overall classifier performance is using a ROC curve analysis. A ROC curve for a classifier plots the False Positive Rate(FPR) on the x-axis and the True Positive Rate(TPR) on the y-axis.

4. RESULT AND ANALYSIS

In our experimental result we found that the accuracy is very high when different classical machine learning algorithms are applied on the original dataset. Although a high TN Rate was achieved, TP Rate is very poor. It indicates that the rate of correctly classification of the negative class (non key phrase) is very high whereas the rate is extremely low for positive class (key phrase). This means that the classifiers are not able to effectively handle the positive instances. But in this experiment our sole intention was to correctly classify the positive instance. The reason behind this unexpected output of the classifier is that the dataset is highly skewed towards the negative class instance. This nature of the dataset is inherent in this experiment because it is intuitive that the number of non key phrases (negative class) is much greater than the number of key phrases (positive class) in a document. Although the learning algorithms have good accuracies, they are not suitable for our experiment as they either fail to predict the positive class instance or do that job with poor performance. This problem is called imbalance learning.

Decision trees do not take all attributes into consideration to form a hypothesis. The inductive bias is to prefer a shorter tree over larger trees. Moreover, like many other learning methods (e.g. rule based), a decision tree searches for a hypotheses from a hypotheses space that would be able to classify all new incoming instances. While doing so, it prefers shorter hypothesis trees over longer once and thus compromises with unique properties of the instances that might lie with an attribute that has not been considered.

Unlike decision tree, k-Nearest Neighbor does not estimate the target function once for the entire instance space, rather does it locally and differently for each new instance to be classified. Also, this method calculates the distance between instances based on all attributes of the instance i.e. on all axes in the Euclidean space containing the instances. This is in contrast to methods such as rule and decision tree that selects a subset of the learning attributes while forming the hypothesis.

Sampling - A very common technique to solve this problem of imbalance learning is sampling. It is the process of balancing the skewed dataset synthetically. There are two types of sampling- oversampling where the minority class instances are increased and under-sampling where the majority class samples are decreased. Note that both under-sampling and oversampling are done taking into account the balancing factor (percentage of minority class in the sample) on a fixed sample size. However both oversampling and under-sampling has some disadvantages. Under-sampling discarded potentially useful data whereas Oversampling increased data by replicating the existing data and let the learning model formulate the rules on the replicated data.

In our experiment we used Synthetic Minority Over-sampling Technique (SMOTE) as described in [15].In this algorithm oversampling is done by generating new instances synthetically rather than replicating the instances. The purpose of sampling is to rebalance a dataset by increasing the number of minority class instances enabling the classifiers to learn more relevant rules on positive instances. However, we note that there is no ideal class distribution. A study done by [16] shows that, given plenty of data when only *n* instances are considered, the optimal distribution generally contains 50% to 90% of the minority class instances. Using the SMOTE algorithm we increased the minority class sample percentage up to 95% with a random interval. At each step we measure the TP rate and TN rate the classifier. We found that the TP rate is increased at a high rate, but simultaneously the TN rate is decreased at a little rate. While we are increasing the minority class sample our aim is not limited to get a high TP rate but also a fairly good TN rate. In Figure 3(right) shows the both the TP rate and TN rate with respect to the minority class samples percentages. Here we found that at around 80-85% the TP rate curve and the TN rate curve is intersecting, which mean that minority class sample percentage 80-85% will give the optimal performance measure. Now we investigated the Area Under the ROC curve (AUC) and the accuracy of the classifier at minority class sample percentage 80-85%.The the AUC is 0.776 and accuracy is 79% as shown in Figure 3(left) and Figure 3(middle). So we used minority class sample percentage as 80-85%(≈83%) further experimental stages. We also performed a comparison between different classifier used in this experiment. The AUC of the J48 is maximum 0.941 as in Figure 4 (left).The TP rate of J48 is 0.92 which is also better than the other two classifiers as plotted in Figure 4 (middle).The accuracy of the classifier is decreased after sampling but the rate of accuracy fall for J48 is lowest (<7%), the accuracy of J48 after sampling is 91.1% (Figure 4 (right)). So out of three classifier we used the overall performance of the J48 Decision Tree is best suitable in our experiment.

5. CONCLUSION

Key phrase extraction is important task in the field of Natural Language Processing and Information Retrieval. It is used in many areas varying from search engines to text categorization. Many research groups are engaged in automatic keyword extraction from documents. In our work we used a supervised machine learning technique along with the linguistic knowledge of the phrases. We used the statistical features *TF X IDF* score and relative position of the phrase and linguistic feature Parts Of Speech tag of the phrase. Our method follows supervised learning by using documents in the training set and extract keywords from the test set from the previous knowledge. We used three different classifier Naïve Bayes',J48 Decision Tree, Instance Based K Nearest Neighbor. The performance of J48 is best among them. We also used sampling technique (SMOTE) to solve the problem balance Learning by increasing the minority class sample and to get higher TP rate and higher TN rate.

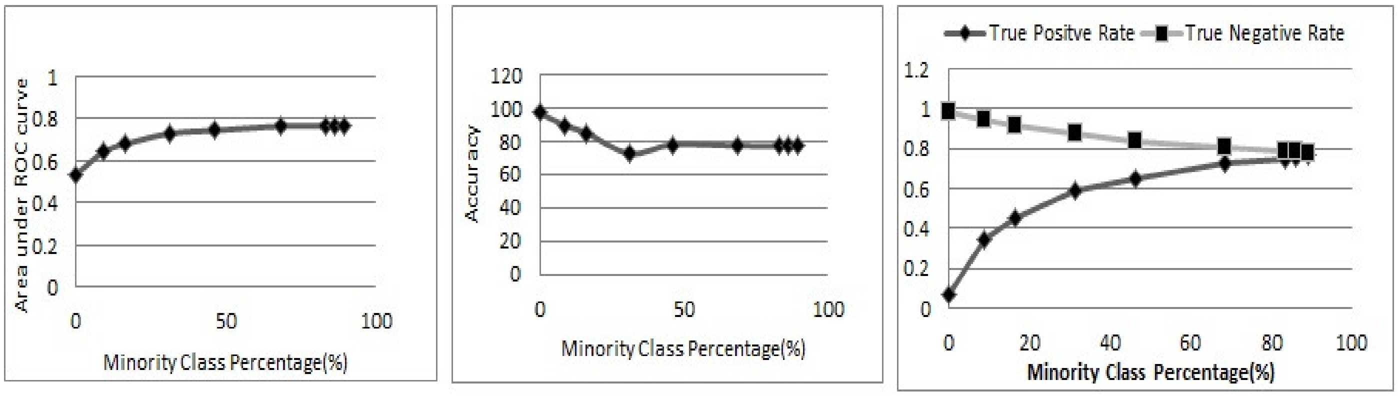

Figure 3:-(a) AUC (left) (b)Accuracy (middle) (c)TP & TN Rate (right) for minority class distribution

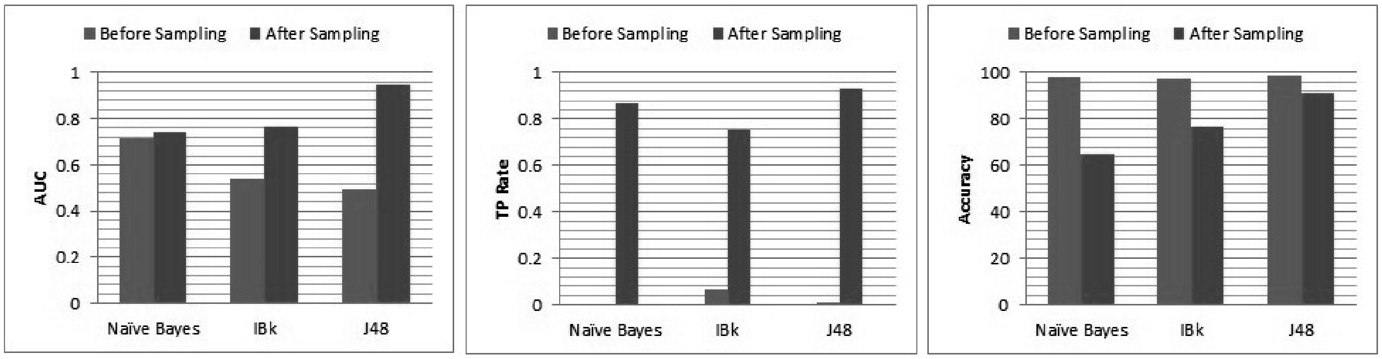

Figure 4:- Comparison of (a) AUC (left) (b)TP Rate (middle) (c)Accuracy (right)

REFERENCES

[1] Krulwich and Burkey .: The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction. IEEE Expert: Intelligent Systems and Their Application. Vol.12 Issue 5 , pp 22 – 27 ,1997 ISSN:0885-9000

[2] Muñoz,A.: Compound key word generation from document databases using a hierarchical clustering ART model. *Intelligent Data Analysis*, 1(1) 1996.

[3] Turney, P. D. : Learning algorithms for keyphrase extraction. *Information Retrieval*, 2: pp 303-336, 2000.

[4] Frank et al.: Domain-specific keyphrase extraction. *Proceedings of IJCAI*, pp. 668-673, 1999.

[5] Hulth, A. : Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of EMNLP. pp.* 216 -223, 2003

[6] Mihalcea, R., and Tarau, P. : TextRank: Bringing order into texts. In *Proceedings of EMNLP*. pp 404-411,2004

[7] Barker, K., and Cornacchia, N. : Using nounphrase heads to extract document keyphrases.Advances in Artificial Intelligence Lecture Notes in Computer Science, 2000, volume 1822/2000, 40-52, 2000

[8] Paice, C., Black, W. : A three-pronged approach to the extraction of key terms and semantic roles. Proceedings of he International Conference on Recent Advances in Natural Language Processing Borovets, Bulgaria, pp 357-363, 2003

[9].Witten et al. KEA: Practical Automatic Keyphrase Extraction. In Proceedings of ACM DL, pp. 254–256, 1999.

[10] Feather, J. and S. P.. International encyclopedia of information and library science. London & New York: Routledge 1996

[11] Silverstein et al. Analysis of a very large Web Search Engine Query Log, ACM SIGIR Forum , 33(3),1999, Originally published as DEC Systems Research Center Technical Note, 1998.

[12] Provost F. Building the Case Against Accuracy Estimation for Comparing Induction Algorithms. Proceedings of the Fifteenth International Conference on Machine Learning, pages 445–453, San Francisco, CA, Morgan Kaufmann 1998.

[13]. Turney .P Extraction of Keyphrases from Text: Evaluation of Four Algorithms, National Research Council, Institute for Information Technology, Technical Report ERB-1051,1997

[14] Manning C. D., Raghaban P., Schutze H. An Introduction to Information Retrieval Cambridge University Press, pp 117 - 120.,2008

[15] Chawla et al. SMOTE: Synthetic Minority Over-sampling Technique, Journal of Aritificial Intelligence Research, pp 321-357, 2002.

[16] Weiss et al . The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical Report ML-TR-44, Rutgers University, Department of Computer Science, 2001.