Available online at www.ijrp.org

IJRP.ORG
International Journal of Research Publications

International Journal of Research Publications
Volume 9 – Issue. 1, July 2018

Text Classification using KNN with different Feature Selection Methods

Rajshree Jodha[1], Gaur Sanjay BC[2], K. R. Chowdhary[3]

[1,2,3] JIET Group of Institutions, Jiet Universe Mogra, Jodhpur(342001), Rajasthan, India

Abstract

This paper presents a fast and efficient approach for text classification using KNN for different feature selection method. Typically, this approach evaluates the performance of the system for minimum number of features required to classify the text documents. 20 Newsgroup dataset collected by Ken Lang, have been taken to check performance of the KNN classifier algorithm. The above dataset is separated into two parts viz. training set (60%) and test set (40%).

The KNN classifier has been implemented against the different number of stemmed and unstemmed features for CHI (Chi-Squared Statistic), IG (Information Gain) and MI (Mutual Information). The Accuracy, Precision, Recall and F1-Score are used to test the system.

Keywords: KNN; Stemmed data; Unstemmed data; Feature extraction; Information Retrieval.

1. INTRODUCTION

Text extraction and classification or categorization are the learning task, where pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labelled documents. Many learning algorithms such as k-nearest neighbor, Support Vector Machines (SVM) [Joachims, T], neural networks [Ozgur et al.], linear least squares fit, and Naive Bayes [McCallum and Nigam] have been applied to text classification.

Availability of expansive number of digital documents from variety of sources including unstructured and semi structured information has given surplus to text mining. The principle undertaking of text Analytics is to empower users to retrieve information and execute function like extraction, classification, summarization and language processing. The above data can be classified into unsupervised, supervised and semi-supervised structures and is helpful to classify and convert the data on logically designed rules. The information i.e data relate to a particular class. This classification can contain a document in two unique labels i.e. single and multi. The previous belongs to one class, whereas the latter belong to many classes.

The Machine Learning and Natural Language Processing are the recent fields for research and implementation of text categorization. It assistances the user to choose from a large number of information as per one's interest. Each of category of document types as document types as unique in itself in terms of its application, understanding and interpretation. Some of algorithms viz. decision tree and rule induction are simpler in comparison to the KNN. But they can work well with smaller size of documents, whereas KNN has capability to classify document of bigger size.

Many authors have proposed many Text categorization methods in the literature but it is very difficult to compare them. Because of the following reasons:

- ➢ Datasets used in the experiments are rarely same in different studies.
- ➢ Some of the authors have worked on the same dataset but different studies usually use different portions of the datasets or they split the datasets as training and test sets differently.

Thus, it is very difficult to compare the results with other's available results in the literature. In this paper, 20 Newsgroup dataset have been taken to check performance of the KNN classifier algorithm.

RELATED WORK

In the last 20 years, content-based document management tasks or Information Retrieval (IR) have gained a prominent status in the information systems field. It is due to the increased availability of documents in digital form and the ensuing need to access them in flexible ways. Text categorization or classification is the activity of labeling natural language texts with thematic categories from a predefined set. In the early '90s, Text categorization become a major subfield of the information systems discipline, thanks to increased applicative interest and to the availability of more powerful hardware. TC is now being applied in many contexts viz. document indexing, document filtering, automated metadata generation etc.

These days many researcher are working on content-based document management tasks or Information Retrieval (IR). Some of the work has been investigated to find gap for this work.

Bijalwen et al. described two tasks of text representation i.e. term weighting and indexing. The primary goal of this paper is to study the effectiveness of different text representation techniques. Weighting is concerned with term frequency (TF) and inverse document frequency (IDF) whereas indexing deals with factual and semantic quality. In this paper, the author conducted experiments to check performance of various representation methods of document viz. TF-IDF, Latent Semantic Indexing (LSI) and multi-word for text representation. As per the experiments conducted, LSI outperforms all other representation methods.

Ankit Basarkar in his thesis work performed a study of different type of vectorizers through which feature vectors, used for document representation and classification, can be generated. Binary, Count and TF-IDF vectorizer techniques were utilized on 20 Newsgroups dataset and their effect on document classification was studied. For each vector representation Naive Bayes classifier was utilized for training and the generated model was tested on test documents. The results were evaluated in two cases, where the first case was when stopwords were removed and in the second one when stopwords were not removed. It was found that TF-IDF performed better in both the cases than Count and Binary vectorizer. Count Vectorizer performed better than Binary vectorizer when the stopwords were removed but lagged behind the latter when stopwords were retained. It was concluded that TF-IDF should be preferred for document representation and classification.

Hakim et al. implemented TF-IDF algorithm for classification. TF-IDF was picked because it counts the word weight by considering recurrence of the word (TF) and in how many files a term can be discovered (IDF). Since the IDF could find in how many files a term can be found, it can control the weight of each word. When a word can be found in so many files, it will be considered as an insignificant word. TF-IDF has been demonstrated in their investigation to make a classifier that could classify news articles in Bahasa Indonesia with a high accuracy i.e. 98.3%.

Saniat et al. in their paper compared general language data classifying techniques using 5 different machine learning algorithms: Naive Bayes, Pegasos, kNN, Perceptron and Rocchio. Additionally, they did some extra comparison of how each type of stemming affects the accuracy of the algorithms. They used Lovins Stemmer, Porter Stemmer and Paice Husk Stemmer and found Paice Husk to be most attractive choice for stemming because it brings down the number of features to the lowest amount.

K-Nearest Neighbors (KNN)

KNN is a Machine Learning Algorithm. In this Classifier each new Document is compared to fundamental documents. This Algorithm checks how a document is classified by looking at only training data that are equal to it. KNN assumes that to divide the documents like a points in the Euclidean area. The distance among the two points of any plane with the p(x,y) and q(a,b) calculated as :

$$d = \sqrt{(x-a)^2 + (y-b)^2} \tag{1}$$

Data Set Preparation: Loading and Pre-Processing

Following are the steps that represent Data Set Preparation:

Dataset Loading: The required dataset, for this thesis 20 Newsgroups dataset, is loaded during execution either directly from internet or from local system on which it is present previously. Case folding is applied to convert all characters into same case, lower case, in order to avoid duplication of words.

Removal of Header/Footers/Quotes: All the documents of the dataset across all categories contain headers/footers/quotes such for example From, Subject, Signature Line etc. These segments need to be eliminated from the actual content in order to avoid overfitting and generate a more generalized classifier model for better classification.

Tokenization: In this each and every document is treated as a string, and then partitioned into tokens containing characters mentioned in condition of regular expression. Tokenization includes separating sequence of strings into phrases, keyword, phrases, words, symbols called tokens. Punctuation marks are not considered in tokenization.

Stop word removal and Stemming: Removing stopwords causes an efficient reduction in the dimensionality of the feature space but we also need to stem words, beginning from other words, so that the dimensionality of the feature space could be decreased to a sensible number. Stemming is a pre-processing step for finding the root morphemes of the words. Let us say, we have the following words, with frequency in a given document;

talk: 5, talking: 6, talked: 3. Instead of considering all the three forms separately, we can consider the root word ‘talk’ with the frequency of 14 since all the three words signify the same meaning in various sense. With the goal to stem the words, we have utilized three stemming algorithms viz. Porter Stemmer, Snowball Stemmer and Lancaster Stemmer. The efficiency of these stemmers is later compared with each other and compared with the un-stemmed documents too.

Feature Representation and Extraction

Vector space model is the most common method used for document representation. Here each document is represented as a vector d and each dimension in the vector corresponds to a distinct term, called as features, in the term space of the document collection. This representation is expressed as:

$$d = (w_1; w_2; ...; w_n) \qquad (2)$$

where w_i denotes weight of term i in document d.

To compute these term weights several methods are formulated. For proper term weighting and feature extraction, we need a method which considers rare terms as similarly as frequent terms, multiple appearances of a term in a document to be more important than single appearances, and is not biased towards long documents. One of the widely used weighting methods taking these properties into account is the term frequency-inverse document frequency (TF-IDF) weighting. Calculation of ‘df’ resembling ‘document frequency’, ‘tf’ resembling ‘term frequency’ and length normalization present in this formula considers the above stated properties simultaneously. Thus, in our work we apply TF-IDF method whose formula is given below:

$$w_{ij} = tf_{ij} \log(N/d_{fi}) \qquad (3)$$

Here, in a document j the weight of a term is i is w_{ij}, frequency of a term i is tf_{ij} in a document j and d_{fij} is the number of documents in which a term i occurs in the whole document collection. N is the whole number of documents. In tf-idf weighting method, if a term often occurs in a document, it is more discriminative whereas if it appears in most of the documents, then it is less discriminative for the content. This constructed vector space model improves the accuracy, efficiency and scalability of the classification model.

Feature Selection

In the text documents, the high dimensionality of features or terms reduces the accuracy of classification due to irrelevant features. Feature selection, other than feature extraction, is one of the well-known method of reducing the dimensionality by removing non-informative words. These irrelevant and misleading words are found by ranking all features according to their importance estimated by a metric and then selecting ones with higher values. The top words extracted out are then used to classify the documents. Hence, to select features from documents Feature selection techniques are used. It aims at reducing time and improving efficiency of classifiers by removing noise features.

Two main policies are used in feature selection viz. global and local policies while in the second one, a different set of features is selected from each class. In global policy, a single set of features is selected from all classes which provide a global view of entire dataset by extracting a single global score from local scores. Thus, it tends to penalize the infrequent classes in highly skewed datasets. In local policy, a different set of features is selected from each class which tends to give equal weightage to each one of them and thus, it optimizes the performance of classification on frequent and infrequent classes.

In this work, global policy of feature selection is considered and implemented three different methods of feature selection. These are chi-squared method, information gain and mutual information, all of which are compared against each other by the accuracy achieved by using them for classification.

Classification

In our work, various supervised machine learning algorithms, viz. kNN (k-Nearest Neighbors), Naive Bayes (Bernoulli and Multinomial Naive Bayes), Perceptron, Random Forest Classifier and SGD (Stochastic Gradient Descent), have been utilized for classification. In classification step, firstly a classifier is generated based on the feature vector obtained from above steps by learning to which class the document belongs. This trained classifier is used to classify unlabeled documents later. These two steps are described below.

Model Generation: It is also called as training phase or learning phase. In this step, a classifier model is generated using many traditional learning algorithms which utilize feature vector. The collection of documents used for this construction is called as Training Set (in our work 60% of dataset is segregated into training set) describing a set of predefined classes. Each document in this set is assumed to belong to a predefined class.

Model Testing or Usage: Also called as testing phase or classification phase, this step is utilized for classification of unlabeled documents using the classifier model. A validation step can also be performed in this phase. After this, label of test document is then matched with classified result to estimate performance measures of a classifier. The set of documents used here is called as Test Set (in our work 40% of dataset is divided into test set) which is independent of Training Set.

RESULTS

The KNN classifier has been implemented against the different number of unstemmed features for CHI, IG and MI. The Accuracy, Precision, Recall and F1-Score are calculated for KNN classifier as shown in Figure 1 and Table 1.

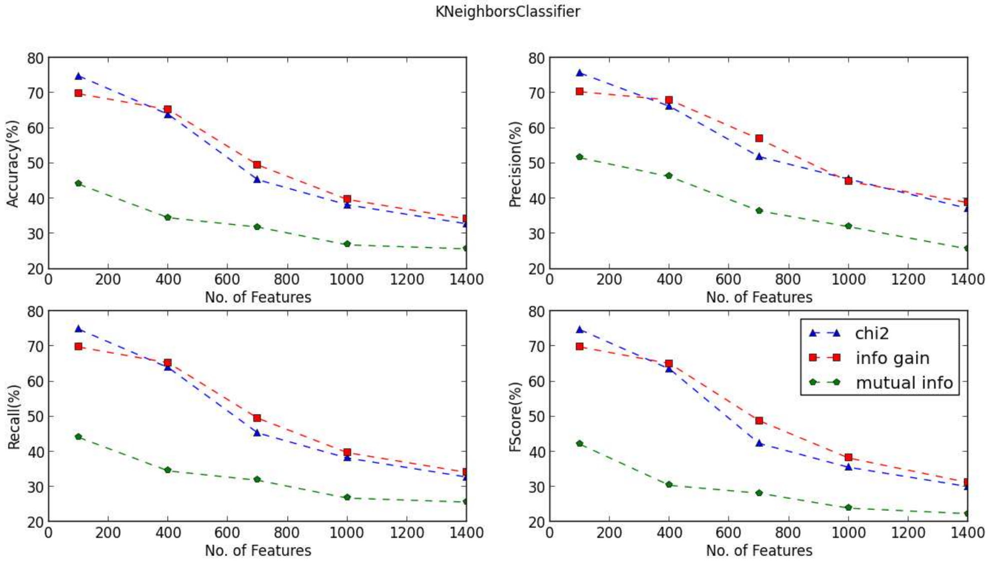

Figure 1: Graphs showing Comparison of Feature-Selection-Methods on Unstemmed Data using KNN Classifier

Table 1: Table showing Comparison of Feature-Selection-Methods on Unstemmed Data using KNN Classifier

No. of Features	Accuracy (in %)			Precision (in %)		
	CHI	IG	MI	CHI	IG	MI
100	74.83	69.77	44.07	75.68	70.38	51.59
400	63.91	65.31	34.55	66.22	68.05	46.23
700	45.34	49.53	31.89	51.90	56.92	36.50
1000	38.22	39.75	26.83	45.53	44.88	32.00
1400	32.82	34.15	25.70	37.17	38.83	25.66
No. of Features	**Recall (in %)**			**F1 Score (in %)**		
	CHI	**IG**	**MI**	**CHI**	**IG**	**MI**
100	74.83	69.77	44.07	74.70	69.79	42.11
400	63.91	65.31	34.55	63.52	65.10	30.45
700	45.34	49.53	31.89	42.37	48.85	28.23
1000	38.22	39.75	26.83	35.59	38.22	23.97
1400	32.82	34.15	25.70	30.10	31.28	22.40

The KNN classifier has been tested against the different number of features stemmed using Lancaster Stemmer for CHI, IG and MI. The Accuracy, Precision, Recall and F1-Score are calculated for KNN classifier as shown in Figure 2 and Table 2.

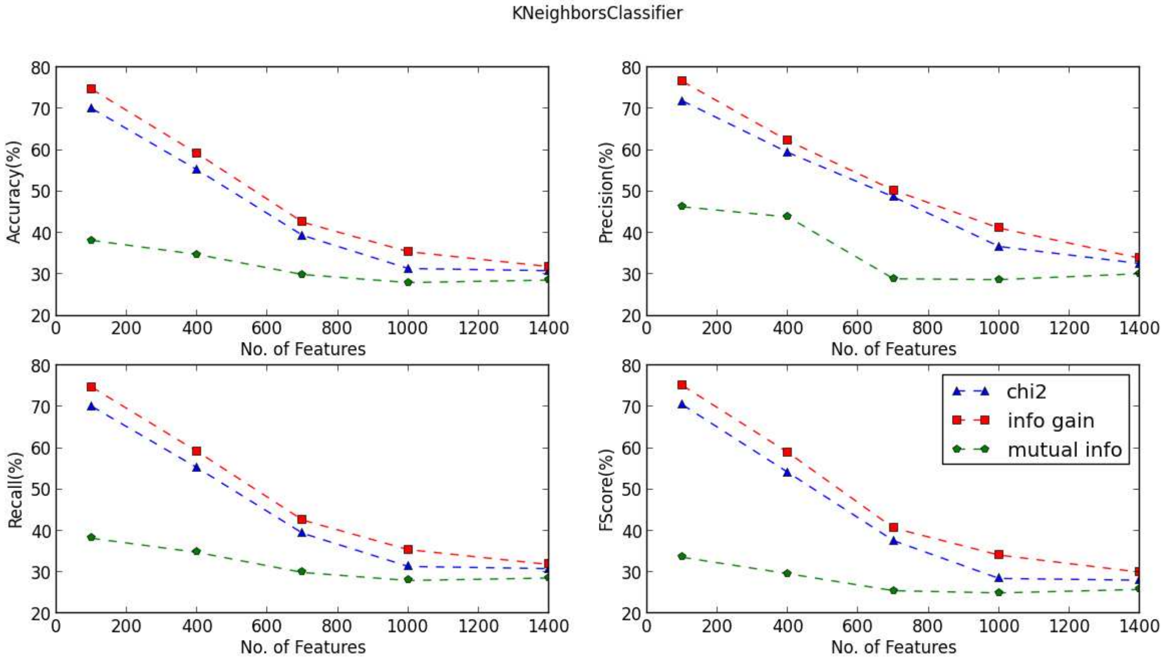

Figure 2: Graphs showing Comparison of Feature-Selection-Methods on Stemmed Data (Lancaster Stemmer) using KNN Classifier

Table 2: Table showing Comparison of Feature-Selection-Methods on Stemmed Data (Lancaster Stemmer) using KNN Classifier

No. of Features	**Accuracy (in %)**			**Precision (in %)**		
	CHI	**IG**	**MI**	**CHI**	**IG**	**MI**
100	70.11	74.47	38.22	71.92	76.63	46.34
400	55.26	59.19	34.82	59.45	62.28	43.88
700	39.41	42.61	29.96	48.72	50.34	28.95
1000	31.42	35.49	28.03	36.75	41.20	28.74
1400	30.89	31.89	28.63	32.63	33.92	30.17

No. of Features	**Recall (in %)**			**F1 Score (in %)**		
	CHI	**IG**	**MI**	**CHI**	**IG**	**MI**
100	70.11	74.47	38.22	70.47	75.14	33.62
400	55.26	59.19	34.82	54.12	58.94	29.67
700	39.41	42.61	29.96	37.65	40.70	25.57
1000	31.42	35.49	28.03	28.52	34.17	25.03
1400	30.89	31.89	28.63	28.10	30.04	25.87

CONCLUSION

This paper proposed a text classifier using KNN for different feature selection method. From the above Results, it is clear that KNN Classifier works better for less number of features. As we increase number of feature number the performance of KNN Classifier decreases. It is due to the following Reasons:

- The first reason is that KNN uses Euclidean distance, which becomes meaningless, when the dimension of the data increases significantly.
- Sometimes incapable to handle more number of features due to confusion between different classes.
- The same experiment has been performed for stemmed data & found that performance of KNN is similar in all cases. Hence in all the above four cases the results of KNN Classifier is almost same. Hence it is suggested that this classifier can give better results for less number of features.

REFERENCES

Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. European Conference on Machine Learning (ECML) (1998)

Ozgur, L., Gungor, T., Gurgen, F.: Adaptive Anti-Spam Filtering for Agglutinative Languages. A Special Case for Turkish, Pattern Recognition Letters, 25 no.16 (2004) p.g. 1819–1831.

McCallum, A., Nigam, K.: A Comparison of Event Models for Nave Bayes Text Classification. Sahami, M. (Ed.), Proc. of AAAI Workshop on Learning for Text Categorization (1998), Madison, WI, p.g. 41–48.

Yang, Y., Liu, X.: A Re-examination of Text Categorization Methods. In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley, US (1996)

Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34 no. 5 (2002), p.g. 1–47.

Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research 3 (2003), p.g. 1289–1305.

Ozgur, A.: Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization. Master's Thesis (2004), Bogazici University, Turkey.

Burges, C. J. C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery Vol. 2 No. 2 (1998) p.g. 121–167.

Vishwanath Bijalwan, Vinay Kumar, Pinki Kumari, Jordan Pascual, "KNN based Machine Learning Approach for Text and Document Mining", International Journal of Database Theory and Applications, Vol.7, No.1, 2014, pp. 61-70.

Ankit Basarkar, "Document Classification using Machine Learning", MS Thesis, San Jose State University, 2017

A. A. Hakim, A. Erwin, K. Eng, M. Galinium, W. Muliady, "Automated document classification for news article in bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach", 6th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 1-4, 2014.

Saniat Javid Sohrawardi, Iftekhar Azam, Shazzad Hosain, "A comparative study of text classification algorithms on user submitted bug reports", 9th International Conference on Digital Information Management (ICDIM), IEEE (2014), pp. 242–247, 2014

Gulden Uchyigit, "Experimental evaluation of feature selection methods for text classification", 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012