Volume 6, Issue 6, June 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Improvement in KNN Classifier (imp-KNN) for Text Categorization

Shaifali Gupta
Student, Deptt. of CSE JMIT Radaur,
Haryana, India

Reena Rani
A.P. Deptt of CSE JMIT, Radaur,
Haryana, India

Abstract: In today's library science, information or computer science, online text classification or text categorization is a huge complication. [1] With the huge growth of online information and data, text categorization has become one of the crucial methods for handling and standardizing text data. Different learning algorithms have been applied on text for categorization. Based on the accuracy and efficiency, KNN (K nearest Neighbour) algorithm prove itself to be very efficient algorithm as compared to any other learning algorithms. The framework of KNN with TF-IDF is studied and some changes need to be done for removing time complexity and improving accuracy so, proposed work is based on using imp-KNN (improved KNN) classifier which is helpful in splitting of training and testing data and take less time from the previous work with KNN algorithm which takes less time and more accuracy and improve text categorization.

Keywords: Document categorization, KNN, imp-KNN, TF-IDF.

I. INTRODUCTION

Very huge growth in the amount of text data leads to development of different automatic methods purposed to increase the speed and efficiency of automated text or document classification with textual content. [1] The documents to be classified can have text, images, music, etc. Each content type requires significant classification methods. Text classification is a technique from the main problems of text mining. Text categorization is defined as the term which helps in assigning uncategorized data or text into fixed or predefined categories. The main objective of text categorization is to assign a category to a new document. Category must be assigned according to their textual content. Document or text can reside in multiple, one or no category.This is based on supervised machine learning method where documents are framed VSM (vector space model) where words are used as important features. First step is to train the data so that when we test the data results must be effective and efficient. Various different types of classification methods have been applied like SVM (support vector machine), Naive Bayesian (NB) classifier, Decision trees, Entropy, Fuzzy logic, KNN (k- nearest-neighbour) etc. KNN performance is quite better than other algorithms but still some improvement is required in KNN for decreasing time complexity and improving the accuracy. KNN is a type of lazy learning algorithm; it is based on finding most similar objects from the sample group with the help of euclidean distance.

II. RELATED WORK

Trstenjaka et al (2013**) [1]** presented the framework of KNN with TF-IDF for text categorization. This framework was totally based on quality and speed of the classification. It helps in finding similar objects based on the euclidean distance and TF-IDF calculates the weight for each term in each document. Both KNN and TF-IDF embedded together prove good together gave good results and confirmed the initial expectations. Framework is performed on different categories of documents and the testing is performed. During testing, classification gives accurate results due to KNN algorithm. This combination gives better results and need to upgrade and need to improve the framework for better and high accuracy results.

Seema Singh et al (2014) **[2]** Text categorization task have gained the attention of researchers in last 10 years with the increase in web-based contents of the documents. For searching a particular document from the web or any large document collection text or document categorization is the most useful task. We demand some better system and enhanced machine learning classifiers to accomplish the task of document categorization. We designed a multi-agent based system which consists of some software hybrid agents that obtains category of a document and interact with each other to take final decision about the category and data is fed to the machine learning classifier in order to enhance the performance.

Hao Lin et al (2014) **[3]** In this paper, we evaluated energy cost of different classifiers and reduced the energy cost by parallelization, trying to find the classifier that performs best on both aspects – effectiveness as well as efficiency.

Rajni Jindal et al (2015) **[4]** This research proposes a novel lexical approach to text categorization in the bio-medical domain. We have proposed a LKNN (Lexical KNN) algorithm, in which lexemes (tokens) are used to represent the medical documents. These tokens are used to classify abstracts by matching them with the standard list of keywords specified as MESH (Medical Subject Headings).This automatically classifies journal articles of medical domain into specific categories. We have used collection of medical documents, called Ohsumed, as the test data for evaluating the proposed approach. The results shows that LKNN outperforms the traditional KNN algorithm in terms of standard F-measure.

Murat Can Ganiz et al (2015) **[5]** Text classification is one of the key methods used in text mining. Basically, traditional classification algorithms from the machine learning fields are used in text classification. These algorithms are primarily designed for structured data. In this paper, we propose a new classifier for textual data, called Supervised Meaning Classifier (SMC). The new SMC classifier use a meaning measure, which is based on Helmholtz principle from Gestalt Theory. In SMC, meaningfulness of the terms in the context of classes are calculated and used for classification of the document.

Pramod Bide et al (2015) **[6]** Searching for similar documents has a crucial role in document management. Because of tremendous increase in documents day by day, it is very essential to segregate these documents in proper clusters. Faster categorization of documents is required in forensic investigation but analysis of these documents is very difficult. So, there is a need to separate multiple collections of documents into similar ones through clustering. Specifying number of clusters is mandatory in existing partitioning algorithms and the output is totally dependent on given input. Over clustering is the major problem in document clustering. The proposed algorithm takes input as Keywords found after extraction and solves the problem of over clustering by dividing documents into small groups using Divide and Conquer Strategy. In this paper, an Improved Document Clustering algorithm is given which generates number of clusters for any text documents andss uses cosine similarity measures to place similar documents in proper clusters. Experimental results showed that accuracy of proposed algorithm is high compared to existing algorithm in terms of F-Measure and time complexity.

III. PROPOSED WORK

This section describes the flow chart of the proposed work. The proposed system can be summarized into three main steps which are integrated to give accurate results: text document representation, classifier construction and performance evaluation.

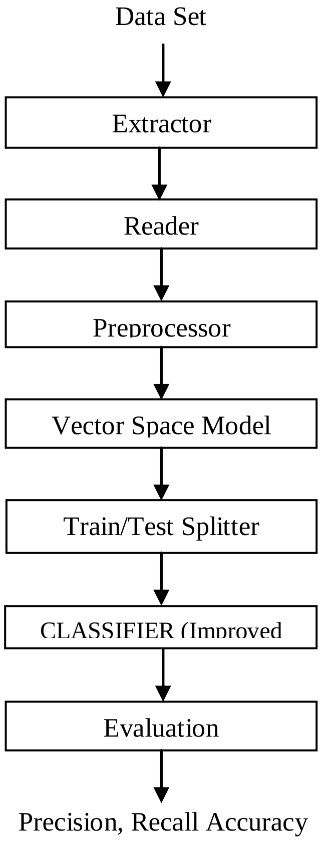

Fig 1. The proposed text categorization system framework

A. Extractor & Reader

Extractor extracts the data set and reads the number of documents, the number of topics and the documents which are related to specific topics. It is an agnostic content summarization technology that automatically parses news, information and documents into relevant and contextually accurate keyword and key phrase summaries. Reader reads input text document and divides the text document into the list of features which are also called (tokens, words, terms or attributes).

B. Preprocessor

Pre-processor processes the document words by removing

- Symbols removal
- Stop words removal
- Lower Case Conversion
- Stemming

The all symbols are removed in pre processing step and a stop list is a list of commonly repeated features which appears in every text document. The common features such as it, he, she and conjunctions such as and, or, but etc. are to be removed because they do not have effect on the categorization process. Stemming is the process of removing affixes (prefixes and suffixes) from the features. It improves the performance of the classifier when the different features are stemmed into a single feature. For example: (convert, converts, converted, and converting) stemming removes different suffixes (s, -ed, -ing) to get a single feature.

C. Vector Sapce Model

In vector space model each input text document can be represented as a vector and each dimension of this space represents a single feature of that vector and based on the frequency of occurrence, weight is assigned to each feature in text document. This representation is known as vector space model. In this step, each feature is assigned to an initial weight equal to 1.

TF-TDF term is used in vector space model for assigning weight to each feature. It determines the relative frequency of the words in a specific document. For calculation, TF-IDF method uses two elements:

TF - term frequency of term in the document (the number of times a term appears in the document)

IDF- inverse document frequency of term i (number of documents where the term appears)

[1]Formula for tf and tdf are:-

$$tf\,(t,d) = 0.5 + \frac{0.5 * f\,(t,\,d)}{\max\{f(w,d) : w \text{ € } d\}}$$

$$idf(t,D) = \log \frac{N}{|\{d \text{ € } D : t \text{ € } d\}|}$$

tfidf(i , d , D) = tf(t , d) * idf(t, D)

For creating VSM (vector space model) :

```
for i = 1 to num docs
    j = 1 to num of unique words

Vsm (i,j)= (tf(i,j) * log 2(num docs/df(j)));
```

j=Number of all unique words

a(0,0)	a(0,1)	a(0,2)	a(0,3)	....	.....	....	....
a(1,0)							
a(2,0)							
a(3,0)							
.....							
.....							
......							

Fig 2. Weight matrix.

D. Train/Test Splitter

In this paper a train/test splitter is used (i.e. random splitter) which only works randomly and the advantage of training the data in this way is during testing the data it gives most accurate results from the previous work done.

E. Document Classfier

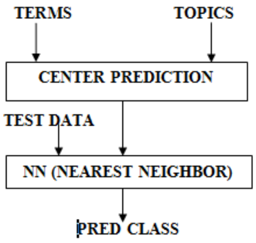

Fig 3. Document classifier

F. Center Prediction

Before calculating the centre prediction, a vocabulary is formed from the documents. Like, for one category a vocabulary of 'n' frequent words are chosen same for category two and same for category three. These 'm'(3*n) words are quite frequent from each category and centre calculation is properly based on the frequent coming words and vocabulary. First 'n' words are centre for one category next 'n' words are centre for second category and next 'n' words are centre for third category. Training of data is based on the centre prediction. Centres are predicted from intelligent vocabulary which contains top rated terms which reduce dimension and complexity and testing become easier.

Vocabulary	Category one n words	Category two n words	Category three n words

Fig 4. Centre prediction

where ‘n’ belongs to number of words taken in respective categories according to dataset.

G. Nearest Neighbour

In this paper centre prediction and nearest neighbour play a major role in proposed work. Testing of data is based on the nearest neighbour. Nearest neighbour is different from KNN (k-nearest-neighbour). KNN works on the principle of calculating centres again and again for each test term but NN works on the principle that it only calculates centre for one time and never update it and it calculates the minimum distance from with the help of Euclidean distance. [1]

$D_{Euclidean}\ (x, y) = \sqrt{(x_1\text{-}y_1)^2 + (x_2\text{-}y_2)^2}$

Where (x_1, x_2) are coordinates of x and (y_1, y_2) are coordinates of y.

IV. EXPERIMENTAL SET UP

The experiments are carried out using mini- newsgroup dataset from UCI KDD Archive which is an online repository of large data set that encompasses a wide variety of data types, analysis tasks and application areas. Mini newsgroup contains 20 groups of 100 documents each. Our experiment uses 3 newsgroups of 100 documents each.

V. RESULTS

In this section, we have investigated the performance of our proposed algorithm iKNN (improved KNN) and compare it with KNN algorithm.

Results	Accuracy	Precision	Recall	F-measure	Time
KNN	0.77	0.67	0.66	0.65	0.015
iKNN	0.91	0.86	0.86	0.85	0.005

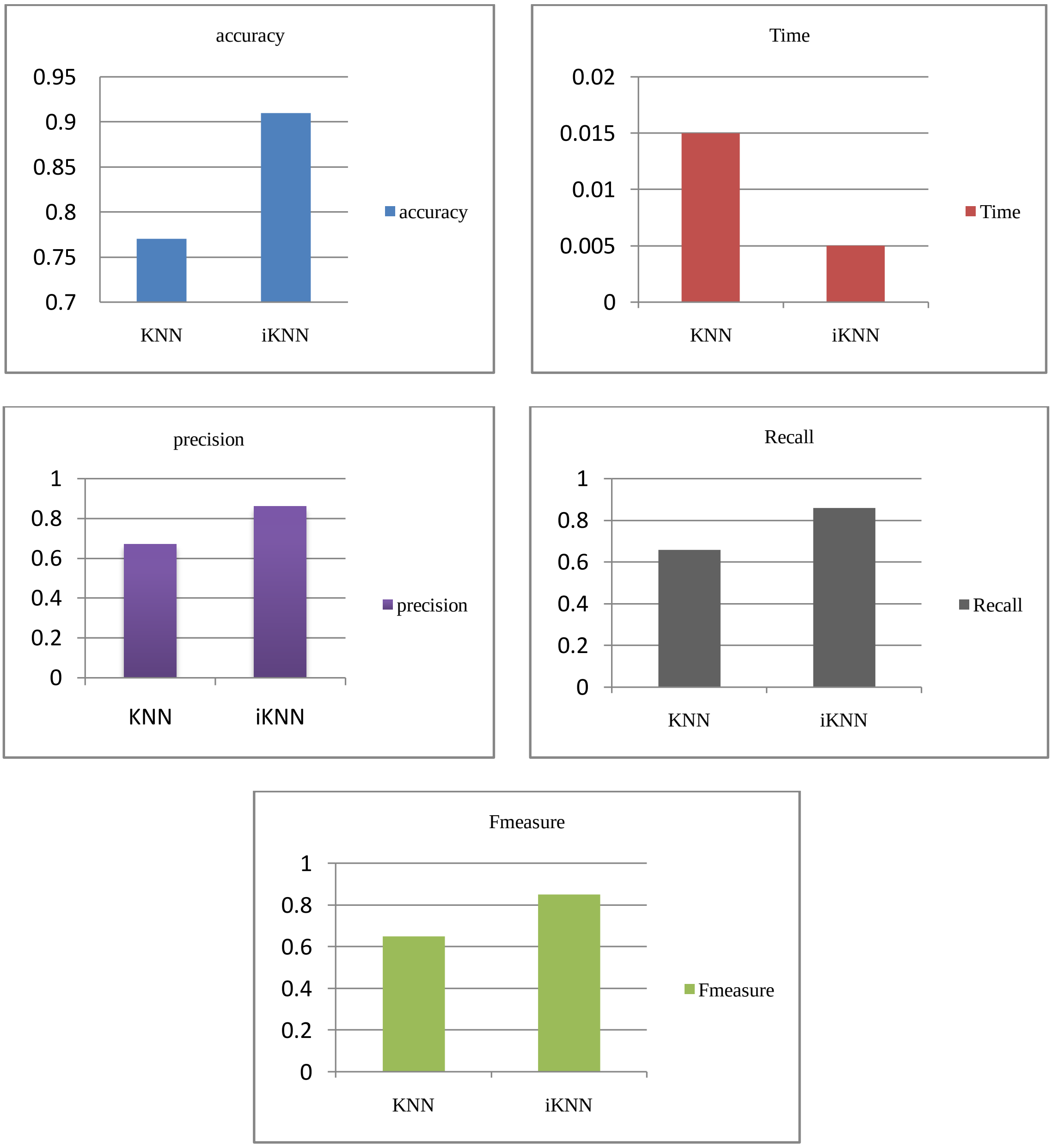

Fig 5. Comparison of both approaches

VI. CONCLUSION AND FUTURE WORK

In this paper we have presented a framework for text classification based on better categorization by using imp-KNN (improved KNN) algorithm instead of KNN. The main motivation for this proposed work is to improve the existing algorithm. Results produced are more efficient and more accurate than existing algorithm. The main factor of time complexity of KNN is reduced with the help of improved algorithm or work. Future work can be proposed for highly accurate results with the help of topic modelling and we can use hybrid models with more effective framework so that results can improve further.

REFERENCES

[1] B. Trstenjak, S. Mikac and D. Donko, “KNN with TF-IDF Based Framework for Text Categorization,” 2014

[2] Seema Singh, Chandra Prakash,”Document Categorization in Multi-Agent Environment with Enhanced Machine Learning Classifier” 2014 IEEE

[3] Hao Lin,”Research on energy-efficient text classification” 2nd International Conference on Information Technology and Electronic Commerce (ICITEC 2014)

[4] Rajni Jindal,Shweta Taneja “A Lexical Approach for Text Categorization of Medical Documents” 2015

[5] Murat Can Ganiz , Melike Tutkan , Selim Akyokus, “A Novel Classifier Based on Meaning for Text Classification” 2015

[6] Pramod Bide,Rajashree Shedge “Improved document clustering using k-means algorithm”