Proceedings of the Second International Conference on Computing Methodologies and Communication (ICCMC 2018)
IEEE Conference Record # 42656; IEEE Xplore ISBN:978-1-5386-3452-3

Evaluating Document Analysis with kNN Based Approaches in Judicial Offices of Bangladesh

Md. Aminul Islam
Department of Computer Science and Engineering
Miltary Institute of Science & Technology
Dhaka,Bangladesh
Email-sumon2907@gmail.com

Md. Jahidul Haque
Department of Mathematics & Physics
North South University
Dhaka, Bangladesh
Email: jahidul.haque@northsouth.edu

Abstract*— In this contemporary era of artificial intelligence, machine learning (ML) algorithms are getting significant attention for the analysis of textual analysis. In recent years, operational improvement in different corporate sectors of Bangladesh are achieved by implementing digitization of the process flow instead of using manual paper trails in offices. Nowadays, judicial sectors are included into sate wide digitalization process by archiving the judiciary records. Despite such improvement, autonomic categorizing of documents using textual analysis is not seen in labeling the correct class of a judicial document. In fact, officers spend lots of time in manual labeling of court related document. In our present investigation, we approached a textual analysis tool that can initiate towards the major solution for solving the manual categorization problem within the judicial sector of Bangladesh. Our objective is to label a normalized text document by implementing ML algorithm into suitable class in terms of the case type. In addition, grammatical analysis of English documents is integrated by the natural language processing (NLP) techniques as well as the filtering of feature sets by TF-IDF based term weighting scheme. The outcomes show the important impacts of NLP techniques for generating useful training data in KNN classification algorithm for the categorization of English documents in Bangladeshi judiciary sector.

Keywords—KNN; judicial document categorization ; natural language processing;TF-IDF; grammatical analysis;text classification;labeling;

I. INTRODUCTION

In the judicial sector, performing the daily judicial activities for the continuous services needs a precise classification of suit related documents. Maintenance of the judicial records is always important and, automation of the process can release the huge pressure at the corporate level of judicial sector in Bangladesh. To perform this requisite task, text classification techniques can be crucial for improving the performance in operational activities with essential backup storage through appropriate categorization. Identifying a document using ML classification alongside the NLP can soothe such processes with accurate categorization of suit documents while analyzing the significant segments of the text document with the linguistic analysis is essential to extract the key textual features. As a consequence, clustering of the textual data will be more accurate through the grammatical analysis and retrieval of the significant terms with the analysis of the sentence clause in different levels from a huge number of text corpora [1]. Labeling the words according to parts of speech is a feasible way of feature extraction which will improve the performance of ML algorithm for document categorization [2].

Text documents are included words, numerals and special characters which can be extracted using different text processing methods of NLP including parts of speech tagging, lemmatization, stop word removal, regularization etc. There are various steps which are essential to ensure the accuracy of classification including data extraction and processing to fit for the classifier as well as for identification of the best classification process [4]. The preprocessed textual data set increases the efficiency of the ML classifier whereas the data preprocessing is included feature extraction and feature selection. The important feature can be extracted through the removal of special characters, repeating words and paragraph indention spaces in the document which are irrelevant terms for the classification algorithm. On the other hand, selection of features includes implementation of various NLP techniques such as parts of speech tagging of the terms, term frequency (TF), inverse document frequency (IDF) and weighting schemes for most important terms within the document corpus. The TF is helpful to determine the most frequent word in a text while IDF detects the documents that are associated with a term [4]. In fact, using the combination of TF and IDF will capacitate the feature selection process more stable than other process such as correlation coefficient process [5].

In this paper, KNN classifier is used to categorize the judicial text document according to the types of suits. TF-IDF based weighting scheme is being used to prepare the feature set of KNN. In evident, Trstenjaka implemented a KNN algorithm for the categorization of text data within TF-IDF based document preprocessing [7]. In the feature selection, we analyzed the grammatical structure of the texts according to the English language for identifying the meaningful terms of the document. Later, TF-IDF based weighting scheme applied for

the most important words of the documents which have higher linguistic impact on the document. A class labeling scheme is also implemented for the feature terms in order to fit the data set for the supervised machine learning algorithm of KNN.

II. Related Works

In the field of text categorization, different ML approaches have been implemented for various domains over the past few decades. To improve the efficiency of the machine learning classifier data representation also plays vital role and for this purpose, analysis of the term as well as the determination of the weight of term are the effective techniques of extraction of the features from a text document[3]. Linguistic analysis is a process for extracting data and in [17] part-of-speech tagging process for the textual data for biomedical domain via token centric method has been introduced. Hung and Chen implemented a sentiment analysis in their research where word disambiguation was solved using different NLP approaches [13]. Different ML classifiers such as K-means, Naive Bayes Classifier [12] and Support vector machine (SVM) has been introduced for text classification. Comparison between the Performances of Bayesian classifier and an decision tree algorithm for categorization of the textual data sets has been presented in [8]. Li and Jain [6] has compared the performance of four different methods for classification and the evaluation result reflected that sometimes a single classifier can perform better than combination of two classifiers. Furthermore, other approaches like artificial neural network (NN) can be used in document classification for its high accuracy regardless its high computational cost [2]. A lexical KNN has been used for the classification for medical data in [15] which has a better performance than traditional KNN. For Weighted K means, every feature is provided a weight which reflects its effect and ability in the document [14].

III. Formulation of the Problem

Text documents in judiciary sector are usually archived as computer-generated document format. Officers categorize each document by observing the title of the document and save the document in homogenous location so that they can easily access the files for future purpose. These computer-generated documents can be categorized using artificial intelligence methodologies. In this paper, we used NLP methods for grammatical analysis of sentences within a document and then used a TF-IDF based weighting scheme for creating feature set of the KNN algorithm. The implementation of the problem is developed using python programming language alongside the python's NLP library NLTK.

A. Feature Extraction

Feature extraction of text includes removing special characters and stop words in a document. Feature extractor outputs a feature vectors having full form of sentence or clause. We used a python program to collect each sentence in a paragraph or table based on the indention. Then a feature extractor program uses to extract the all the words from the document including all special character. After that feature extractor filters all the special characters from the token list. The new token list contains the words. Then the program calculates the parts of speech of each token. Parts of speech tagging for each token are used for the feature selection method where every token is analyzed in terms of their position and impact in the sentence using the feature selection module presented in Fig. 1.and new terms are created for the ML dataset.

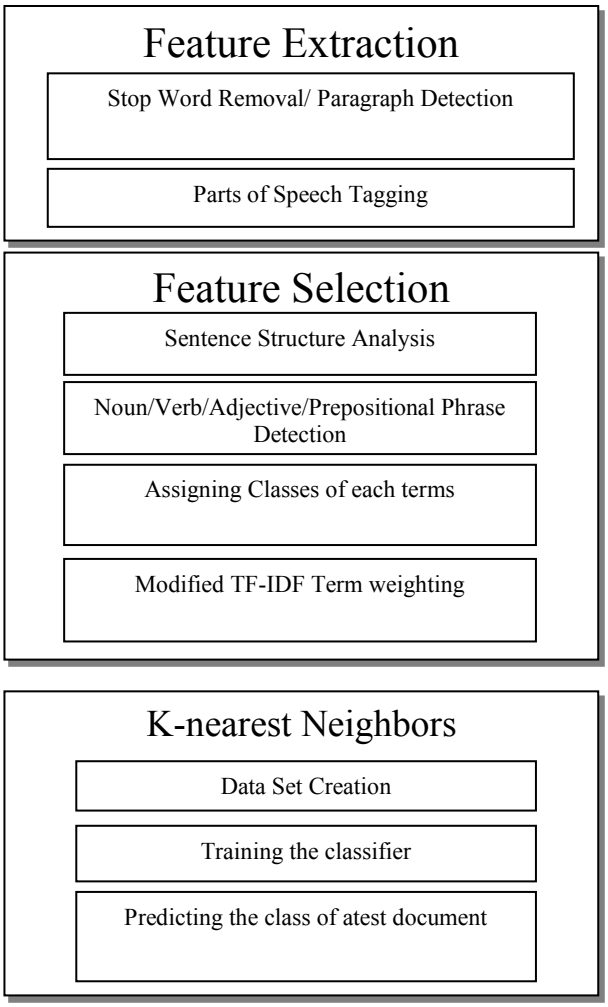

Figure 1: The Implementation stages of the document classification

B. Feature Selection

Feature Selection method is the dimension reduction method for reducing the unnecessary terms within the

document to create better data set for the ML classifier. The feature selection has two major parts in this implementation:

a)Linguistic analysis: Linguistic analysis finds the proper meaning of the terms within a document. The system finds the words each sentence and then create terms based on the linguistic point of view. Different NLP technique such as part of speech (POS) tagging and chunking can be implemented to analyze the data at [9] whereas in [10] N-gram representation and similarity of the longest common chunk has been represented to analysis the common feature between the text datasets.The most important feature among the words in a sentence are the subject and predicate. Subject mainly consists the Noun phrase and predicate contains verbal phrase, adjective phrase and preposition phrase. In [11] noun phrase chunking with filtering process has been implemented to analyze the text data and we also analyzed clause from the sentences with the tokenization process for the terms.Subjects are the most important term in the document vector and thus hold highest preference among other phrases. Fig. 2 presents the output of the program that identifies the phrase which are :

1. Noun Phrase (NounPhrase)
2. Verbal Phrase (VerbPhrase)
3. Adjective Phrase (AdjectivePhrase)
4. Prepositional Phrase (PPhrase)

The preference of the phrase terms can be calculated using the of preference order where the phrase with the highest length is given to the highest value. The linguistic weight of a term (t) in a k[th] document can be computed with the following equation:

$$Lw_{t_k} = (lenght * pw_i), \quad i \in \{NounPhrase \geq 1, VerbPhrase \geq 1, AdjectivePhrase \geq 1, PPhrase \geq 1,\} \tag{1}$$

$$pw_i = \begin{cases} 1 & if\ i = NounPhraase \\ 0.5 & if\ i = VerbPhrase \\ 0.3 & if\ i = AdjectivePhrase \\ 0.2 & if\ i = PPhrase \end{cases}$$

*b) Modified TF-IDF weighting scheme:*After the setting the linguistic weight of the terms, a term weighting scheme is applied in the set of all the terms using a modified TF-IDF approach of the terms where the linguistic feature value is taken into consideration. For a term t in a document d in document set D, the term frequency is calculated as follows:

$$T_{f(t,d)} = \sum_{i=0}^{T_n} f_{(t_i,d)}\, W_{(t_i,d)} \tag{2}$$

$$W_{t_k} = Lw_{t_k} D_k \tag{3}$$

Here, W_{t_k} is the weight of the term t in k[th]document where D_k is the document relevance in the whole document set D. The equation for the document relevance can be computed as follows:

$$D_k = \frac{\sum_{i=1, i\neq k}^{N} Sim(\vec{d}_k, \vec{d}_i)}{N-1}, \quad \text{i}\in D\text{; N=Number of total documents} \tag{4}$$

The frequency of the term denotes the term TF and TF-IDF is a process which is used to provide weight to each term to denote how much effect of the term reflects in the text[7].We used here the product of TF (Term frequency) and IDF (Inverse document frequency) to provide weight each term and select the important term to ease the performance of the classifier.Again, the Inverse document frequency of the terms can be computed using (5) which is also used at [16]. In addition, the combined weight for a term t using both TF and IDF are as follows for this problem domain:

$$DF_t = \sum_{i=0}^{N} \begin{cases} 1 & t \in d \\ 0 & otherwise \end{cases} \tag{5}$$

$$(IDF)_t = log\ (N/DF_t)\ +\ 1 \tag{6}$$

$$T_{(f,idf)} = T_{f(t,d)} . (IDF)_t \tag{7}$$

a) Labeling the feature terms: The collected feature sets are then labeled with the normalized term weight $T_{(f,idf)}$. The labels of the collected set are termed as the class vector of the ML algortihm. The labels are valued at the normalized term weight $T_{(f,idf)}$.Some instances of feature terms in a document after processing is represented in Fig. 3.

Feature Name	Feature class
'the criminal case'	'criminal', 'case', 'section',
'submitted in police station'	'Investigation ', 'section', 'case',
'section 154'	'Information', 'section', 'case',
Date of submission	'date', 'informer', 'case',
Place of incident	'criminal', 'police', 'case',
Date of report sent	'informer', 'submitted', 'report',
'Union-khajnogor'	'information', 'district', 'case',

Figure 3: Instances of the Feature sets in the document vector.

C. ML Classification

The Machine learning algorithms are essential for categorization of any feature vector. In our present analysis, we used K- nearest neighbor algorithm for implementing the classification of the English judiciary documents. For KNN, feature vector $F_i = (F_1, F_1, F_2, F_3, \ldots\ldots, F_N)$is assigned to a class vector $C_i = (C_1, C_2, C_3, \ldots\ldots, C_N)$ to classify a document D. For a given value of K, the algorithm finds its nearest neighbors having similar weight values.

IV. Results and Discussion

The implementation of the document classification includes the measuring the feature set weights and labeling the features to new classes for feeding the ML classifier. In this project, NLTK library is used for using NLP techniques. In addition, scikit-learn library is also used for implementing classification model.

In Data preprocessing stage, grammatical structure analysis is performed on raw data of the document presented in Fig.2. Then the words are grouped together into terms to fit in the ML classifier. Some of the instances of the phrases are given in Fig 3.

Label	Output
Sentence A with POS tagging after removing special characters	[('Information', 'NN'), ('for', 'IN'), ('the', 'DT'), ('criminal', 'JJ'), ('case', 'NN'), ('under', 'IN'), ('section', 'NN'), ('154', 'CD'), ('submitted', 'VBN'), ('in', 'IN'), ('police', 'NN'), ('station', 'NN')]
Sentence A after Phrasing	[([('Information', 'NN')], 'NounPhrase'),([('the', 'DT'), ('criminal', 'JJ'), ('case', 'N N')], 'NounPhrase'),([('submitted', 'VBN '), ('in', 'IN'), ('police', 'NN'), ('station', 'N N'), 'VerbPhrase']]
Sentence A phrases with phraseLength = 1	[([('Information', 'NN')], 'NounPhrase')]
Sentence A phrases with phraseLength> 1	[[('the', 'DT'), ('criminal', 'JJ'),('case','NN')],'NounPhrase')] [('section', 'NN'),('154', 'CD')], 'NounPhrase'] [('submitted', 'VBN'),('in', 'IN'), ('police', 'NN'),('station', 'NN'), 'VerbPhrase']

Figure 2: Identifying different types of phrases

After determining the terms, the weight of each term is calculated in (7). The calculated weight and classes of each term are then transformed to a normalized vector to use as the dataset of ML classifier. The accuracy of the KNN algorithm is calculated of both the weighted data set and un-weighted dataset. Table 1 shows the comparisons of some of the data sets of UCI dataset repository [18] and present data set.

Data Name	Normal KNN	KNN(Present)
Eco-Hotel	0.04092	0.03488
Legal Case Reports Data Set	0.25714	0.22456
Present	0.30335	0.25345

Table 2: Comparison of datasets for the KNN algorithm.

V. Conclusion

The importance of data preprocessing in document classification is very vital for getting good classification accuracy which are presented in results section. But, unfortunately KNN often fails to classify at higher accuracy. In order to get more good results other machine learning approaches like support vector machine (SVM), Naive Bayes Classifier can be used in context of term weighting scheme for the determination of feature set creation. Court-related document can be easily categorized using our proposed method to enhance the operational efficiency in the judiciary sector of Bangladesh.

Acknowledgment

The authors gratefully acknowledge Wali Mohammad Abdullah, Military Institute of Science & Technology, Bangladesh for his valuable suggestion in this research project.

Future Work

In this paper ML classifier is implemented to classify the English documents of the judicial system of Bangladesh. Our future work is to use ML classifier to classify the judicial documents of Bangladesh which are written in the Bengali language. Furthermore, future work will include the analysis of the performance of different classifiers

References

[1] Thaoroijam, Kabita. "A Study on Document Classification using Machine Learning Techniques." International Journal of Computer Science Issues (IJCSI) 11.2 (2014): 217.

[2] Khan, Aurangzeb, et al. "A review of machine learning algorithms for text-documents classification." Journal of advances in information technology 1.1 (2010): 4-20.

[3] Sebastiani, Fabrizio. "Machine learning in automated text categorization." ACM computing surveys (CSUR) 34.1 (2002): 1-47.

[4] Brücher, Heide, Gerhard Knolmayer, and Marc-André Mittermayer. "Document classification methods for organizing explicit knowledge." (2002).

[5] Wei, Chih-Ping, and Yuan-Xin Dong. "A mining-based category evolution approach to managing online document categories." System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on. IEEE, 2001.

[6] Li, Yong H., and Anil K. Jain. "Classification of text documents." The Computer Journal 41.8 (1998): 537-546.

[7] Trstenjak, Bruno, Sasa Mikac, and Dzenana Donko. "KNN with TF-IDF based Framework for Text Categorization." Procedia Engineering 69 (2014): 1356-1364.

[8] Lewis, David D., and Marc Ringuette. "A comparison of two learning algorithms for text categorization." Third annual symposium on document analysis and information retrieval. Vol. 33. 1994.

[9] Rafiei, Javad, et al. "Source Retrieval Plagiarism Detection based on Noun Phrase and Keyword Phrase Extraction."

[10] Sánchez-Vega, Fernando, et al. "Determining and characterizing the reused text for plagiarism detection." Expert Systems with Applications 40.5 (2013): 1804-1813.

[11] Bui, Duy Duc An, Guilherme Del Fiol, and Siddhartha Jonnalagadda. "PDF text classification to leverage information extraction from publication reports." Journal of biomedical informatics 61 (2016): 141-148.

[12] Chen, Jingnian, et al. "Feature selection for text classification with Naïve Bayes." Expert Systems with Applications 36.3 (2009): 5432-5435.

[13] Hung, Chihli, and Shiuan-Jeng Chen. "Word sense disambiguation based sentiment lexicons for sentiment classification." Knowledge-Based Systems 110 (2016): 224-232.

[14] Gao, Yunlong, and Feng Gao. "Edited AdaBoost by weighted kNN." Neurocomputing 73.16 (2010): 3079-3088.

[15] Jindal, Rajni, and Shweta Taneja. "A Lexical Approach for Text Categorization of Medical Documents." Procedia Computer Science 46 (2015): 314-320.

[16] Sabbah, Thabit, et al. "Modified frequency-based term weighting schemes for text classification." Applied Soft Computing 58 (2017): 193-206.

[17] Barrett, Neil, and Jens Weber-Jahnke. "A token centric part-of-speech tagger for biomedical text." Artificial intelligence in medicine 61.1 (2014): 11-20.

[18] [1]"UCI Machine Learning Repository: Data Sets", Archive.ics.uci.edu, 2017.