IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

Legal Document Classification using TF-IDF and KNN

Mr. Nikhil Wani, Ms. Gayatri Mangire, Mr. Aman Kumar, Ms. Nandini Solse, Mrs. P. S. Gaikwad
Department of Computer Engineering
AISSMS-Institute of Information Technology, Pune, Maharashtra, India

Abstract: *The increase in use of NLP, throughout various domains has travelled its way upto legal environment in the recent years. The collection and analysis of data required in lawsuits and topic segmentation of the filed petitions into the respective categories electronically can reduce a significant amount of time and cost compared to human efforts. Our aim is to come up with the process of automating the Text Classification of documents. We plan to implement NLP techniques such as count Vectorizer along with TF-IDF and KNN to categorize the legal documents in a supervised environment. We have proposed a model which works on high dimensional data.*

Keywords: Text Classification, Countvectorizer, TFIDF (Term Frequency–Inverse Document Frequency), KNN.

I. INTRODUCTION

Today, the overwhelming number of lawsuits filed at the court has caused a strain on the Judicial case management system. Manually organizing the case documents and classifying them into respective categories requires experts in the judicial system and highly trained readers. Currently the judicial system manually organizes the case documents according to the field of law. Also this requires a significant amount of time and costs. Delay of ruling or inaccurate ruling and other failures in judicial systems have persisted due to poor management of the lawsuits.

If the documents are classified electronically using machine learning algorithms they can be classified seamlessly. We can group them into right categories by comparing similar aspects of previously filed cases and help in providing justice to every case with an immaculate judgment. Text Classification is the process of categorizing a document into some set of already classified documents by recognising the patterns and its context.

This process can be executed under a supervised or unsupervised environment, it includes the tasks of collection of documents, pre-processing and classification. In this paper we have conducted the supervised training of data and classification of legal documents based on a pre-categorised dataset. We have split the available data into training and testing data.

In this paper, we have made an attempt to focus on the need of automating the process of grouping of documents into two broad categories namely: Civil and Criminal. The initial phase of the process involves noise reduction of a particular document i.e. tokenization of text and stop-word removal along with lemmatization of the corpus. In the later phase. These steps help in the immaculate classification of documents into their respective domains.

II. LITERATURE SURVEY

Herbert L Roitblat et al. [1] has conducted a study to compare an original categorization with automatic categorization. These systems were provided by legal service providers. The original categorization was obtained through the Department of Justice Request, which produced multiple attorneys manually applying index terms, i.e. they would have to read the documents, determine what they were about, and categorize them. A total of 225 attorneys were included in the batch of manual categorization team. This experiment concludes that the automatic categorization is efficient relative to the manual categorization.

Peter Gronvall et al. [2] have explained a new approach of predictive coding during the classification of legal documents. This paper proposes an explainable predictive coding approach along with simple explainable predictive modeling rather than a text classification approach. This approach can locate a responsive snippet within a document. They have included actual legal documents in this research. The first phase consists of identifying responsive

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

documents after which the rationale model is trained to generate one or more rationale in the later phase. The authors believe that there is significant potential in explainable AI and can be used in classification of legal documents.

Mariana Y. Noguti [3] and other members have discussed the use of NLP technique for textual classification. Their aim in this paper is to categorize the description of services which are provided by the Prosecutors Office of the State of Parana to the population. The have classified and assigned different petitions to their respective field of law. This resulted in reduction of time and cost of the manual assignment of these documents. They have also evaluated three different models namely, linear model, boosted tree, neural network. Using the combination of word2vec and RNN lead to the accuracy of 90% and f1-score of 85%. Thus they propose to improve the workflow of the prosecutive office.

Jack R. Brzezinski et.al [4] focuses on machine learning algorithms to classify documents for information retrieval . They have applied logistic regression of three different types namely, binary classification, multiple classification, and hierarchical classification. As they have applied supervised learning similar to us, each document belongs to a given class. Documents belonging to the same class are semantically similar. They have performed two sets of trials using term matrix and singular value decomposition.

Yong-Quan Yang et. al [5] has explained the need to filter out information from a large amount of text. This paper shows a vector representation of feature words based on the deep learning tool Word2vec, and the weight of the featured words is calculated by using an improved TF-IDF algorithm. By multiplying the weight of the featured word and the word vector, the vector representation of the featured word is realized. To overcome disadvantages of the high-dimensional sparse features in the traditional vector space model and the incompatibility between the terms, the Word2vec-based Text Vector Space Model is introduced for Text Representation and the text is trained and classified by the weighted word2vec model.

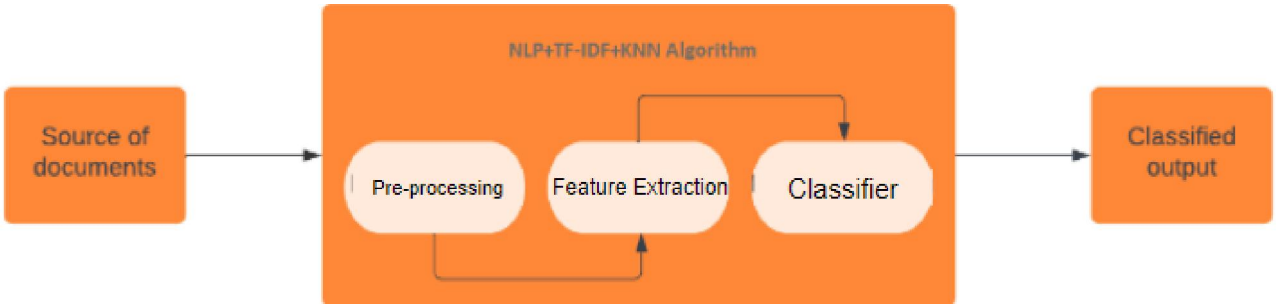
Pascal Soucy et. al [6] has explained Simple KNN Algorithm for Text Categorization. For text categorization this paper proposes to use a simple KNN algorithm for non-weighted features. They propose to use a method for implementing feature interaction in which interdependent words are analyzed. This will reduce the considerable amount of selected features, making KNN algorithm applicable in contexts where both the volume of documents and the volume of the vocabulary are high, like with the WWW (World Wide Web). Hence the proposed algorithm is efficient in terms of predictability and interpretability.

Dongyuan Wang et. al [7] has explained the effect of KNN(K-Nearest Neighbor) classifier. This paper proposes an improved KNN algorithm, which can calculate similarity by considering the interaction, coupling relationship between the internal document and the document. This paper calculates the similarity between the text to be measured and the training text by coupling the similarities. Whole process is divided into 4 major tasks. They are as follows: 1) Text representation and Preprocessing 2) Feature selection 3) Feature weight 4) Evaluation index. Purpose of text classification proposed by this paper is to make the accuracy and recall rate as high as possible, while the F1 value is to find a balance point between the two points

Rosnafisah Bte. Sulaiman et. al [8] have demonstrated the use of KNN Algorithm for Classification of Textual Documents. IN this paper, they have implemented the K-Nearest Neighbors (KNN) algorithm using R language. The experiment performed was put forth to challenge the KNN algorithm to find the proper value of k which represents the number of neighbors plotted. When the value of k is taken between 1 to 50 the machine learning system shows its best result. Although, if the value of k is increased above 50 the accuracy falls sharply..

III. SYSTEM ARCHITECTURE

The system architecture consists of 3 phases. Below is the diagram showing the system architecture.



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

IV. PROPOSED METHOD

4.1 Database Creation

The process starts from combining all the legal documents into a csv format file. In the csv format file, a single row contains document information. The pdf consists of a total of more than one lakh words.

4.2 Preprocessing

The first step in pre-processing consists of compilation of multiple documents into a csv format file, since csv format is suitable for processing of the documents. Next, the text undergoes processes of data cleansing, in which the words are tokenized ie. where the sentences are divided into multiple words. It is then standardized by converting all the words to lowercase. After Punctuation removal, Parts of Speech tagging and removal of Numeric data, the texts undergo the most essential step called as lemmetizations. In lemmatization the words are morphologically analyzed to remove inflectional endings and return the base words, eg. playing, played words change to play. Lemmatization and PoS tagging is based on the open-space library called spaCY.

Word	Stemming	Lemmatization
production	product	production
Computers	comput	computer
populated	popul	populated

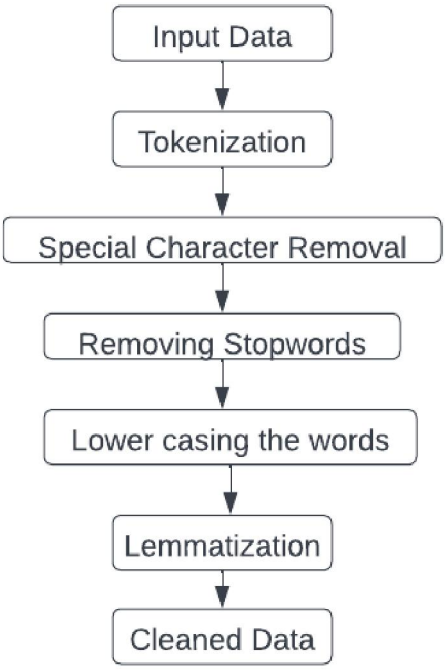


Figure: Data Preprocessing.

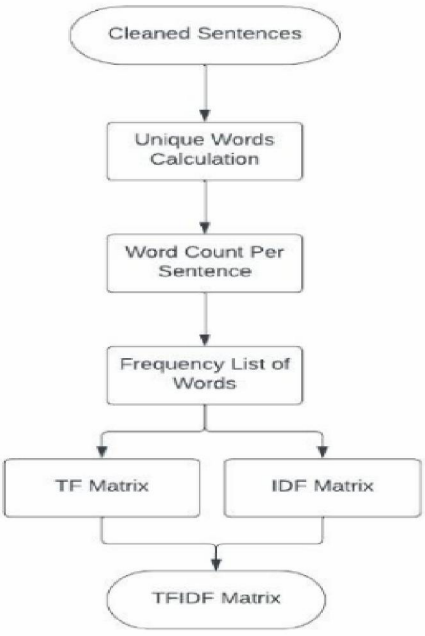
4.3 Feature Extraction

Selecting a subset of important and domain related words from the corpus of the document is called as feature selection. Text categorization may involve hundreds of thousands of features, most of them being irrelevant. During the classification, relevant features are hard to define since there may exist feature interactions that keep the relevant features from being identified.

The earlier step of pre-processing reduces the variability of the terms in the document. Tf means term-frequency while tf-idf means term-frequency times inverse document-frequency. Tf-idf transform, found to be useful in document

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

classification, is used to find the weights of the words after the document is pre-processed. Instead of using raw frequency, ti-idf is used to find the occurrence of words in a document and reduce the impact of words which occur frequently in a given corpus.



TF-IDF is the product of Term Frequency and Inverse Document Frequency.
Term Frequency= Number of times a term(t) appears in a document / Total number of words in the document.
Inverse document Frequency=log(Total number of document(N)/Number of documents which has the term(t) in it).

$$\mathbf{idf(t) = log\ [\ n\ /\ df(t)\ ] + 1}$$

The formula used for calculating tf-idf is as follows-

$$\text{tf-idf(t, d)} = \text{tf(t, d)} * \text{idf(t)}$$

4.4 Classifier

Theoretically, each document is represented as a vector as below:

<w1(x),w2(x),w3(x),w4(x)...wf(x)> []

This where x is a document and wj(x) is the weight of the jth term. The weight here is set according to the output of the tfidf vectorizer.

In KNN algorithm, distance between the test object and its neighbor is used as a basis to weight the contribution of each k neighbor in the class assignment process. The accuracy of the document d belonging to the class c is calculated as follows:

$$\text{Confidence}(c,d) = \frac{\sum_{k_i' \in K | (Class(k_i')=c)} Sim(k_i', d)}{\sum_{k_i \in K} Sim(k_i, d)}$$

where sim is a value of similarity function used to compare a particular document with its neighbors.

To compare document d with instance i, we choose the CosSim function which is particularly simple using our binary term weight approach :

$$\text{CosSim}(i,d) = \frac{C}{\sqrt{A * B}}$$

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

where C is the number of terms that i and d share, A is the number of terms in i and B the number of terms in d. The neighbor set K of d thus comprises the k different instances that rank the top according to that measure..

V. RESULTS AND DISCUSSION

5.1 Dataset

The dataset used for classification purpose is a set of documents which we obtained from the website of the Indian Supreme Court. The dataset contains 500 documents. Later we split this data into a training and testing dataset. We trained the model with the training data set and tested it on the remaining documents.

5.2 Result

After training this model, we got an accuracy of 99%. Below is the confusion matrix created using the test dataset.

```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(Y,predicted)
print(cm)
```

```
[[50  0]
 [ 1 49]]
```

By the above matric we conclude that:
True Positive = 50
True Negative = 49
False Positive = 0
False Negative = 1
By using the below formula we get the accuracy of the model

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy= 99%

The reason for this high level of accuracy is that we have used high dimensional data but sparse dataset.
We use the formula given below to get the precision:

$$Precision = \frac{TP}{(TP + FP)}$$

Precision = 1

For recall we use the formula below:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Recall = 0.98

To cross validate the above accuracy we use numpy library to calculate the accuracy as follows:

```
import numpy as np
print('We got an accuracy of',np.mean(predicted == Y)*100, '% over the test data.')
```

```
We got an accuracy of 99.0 % over the test data.
```

We use the formula below for calculating the F1 score:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

F1 = 0.99

Nowadays, most of the models are based on models (BERT - Bidirectional Encoder Representations from Transformers, DistilBERT, and RoBERTa). The above models work well for short passages and questions. But we have proposed a model which gives better accuracy for long text. In one document we have more than 1,00,000 words still it gives 99%accuracy. Since we have used high dimensional data sets, the accuracy has increased considerably.

ACCURACY	99%
PRECISION	1
RECALL	0.98
F1 SCORE	0.99

VI. CONCLUSION AND FUTURE WORK

The aim of this project is to create an automatic system for categorizing the legal data into relevant domains of law. This results in faster document distribution and requires minimum manual effort along with saving of costs. This project can further be used by lawyers with a huge number of cases to study and categorize into different domains.

In future works we intend to explore deep into the study of classification of documents and build a model capable of finding the relevant words in a particular document and include multi-class labeling of data.

REFERENCES

[1]. Herbert L Roitblat, Anne Kershaw and Patrick Oot, “Document Categorization in Legal Electronic Discovery: Computer Classification vs. Manual Review”, Journal of the American Society for Information Science and Technology Volume 61Issue 1January 2010 pp 70–80.

[2]. R. Chhatwal, P. Gronvall, N. Huber-Fliflet, R. Keeling, J. Zhang and H. Zhao, "Explainable Text Classification in Legal Document Review A Case Study of Explainable Predictive Coding," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 1905-1911, doi: 10.1109/BigData.2018.8622073

[3]. R. Beale M. Y. Noguti, E. Vellasques and L. S. Oliveira, "Legal Document Classification: An Application to Law Area Prediction of Petitions to Public Prosecution Service," 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207211.

[4]. J. R. Brzezinski and G. J. Knafl, "Logistic regression modeling for context-based classification," Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99, 1999, pp. 755-759, doi: 10.1109/DEXA.1999.795279.

[5]. C. Liu, Y. Sheng, Z. Wei and Y. Yang, "Research of Text Classification Based on Improved TF-IDF Algorithm," 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), 2018, pp. 218-222, doi: 10.1109/IRCE.2018.8492945.

[6]. P. Soucy and G. W. Mineau, "A simple KNN algorithm for text categorization," Proceedings 2001 IEEE International Conference on Data Mining, 2001, pp. 647-648, doi: 10.1109/ICDM.2001.989592.

[7]. H. Li, H. Jiang, D. Wang and B. Han, "An Improved KNN Algorithm for Text Classification," 2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), 2018, pp. 1081-1085, doi: 10.1109/IMCCC.2018.00225.

[8]. A. Moldagulova and R. B. Sulaiman, "Using KNN algorithm for classification of textual documents," 2017 8th International Conference on Information Technology (ICIT), 2017, pp. 665-671, doi: 10.1109/ICITECH.2017.8079924.