AUTOMATED PATENT CLASSIFICATION USING MACHINE LEARNING

HEMESH TALLURI

Supervisor: Diego Moll´a-Aliod

ITEC810 Information Technology Project Final Report

For the degree of

Master of IT in Internetworking and Cybersecurity

At Department of Computing, Macquarie University

Sydney, Australia

10th November 2018.

TABLE OF CONTENTS

ABSTRACT:

The rapid increase in the patent applications in the past few decades has raised the alarm bell for developing sophisticated tools for analyzing the patent documents that classify the documents, forecast trends in the technology, identify technology hotspots and that detect any infringement of patents. The application of machine learning techniques to this corpus may help relieve the need for highly trained human classifiers. However, some of the questions that arise are the basis of classification, the choosing of an algorithm which gives better performance and accuracy. Text Classification consists of a set of phases, each phase can be accomplished using various statistical machine learning techniques. Selecting the proper statistical technique that should be used in each phase affects the efficiency of text classification performance. The main objective of this project is to design a tool which enables the automated categorization of patent documents using suitable machine learning algorithm under the sections of International Patent Classification System (IPC). For this purpose, a classification model is to be developed using machine learning techniques by extracting the useful features from the documents such that it accurately classifies the documents. The classification of documents is based on the features that are extracted from them. The outcome of this project is to provide the tool which can predict the unlabelled documents by finding the appropriate classification model.

SECTION 1: PROJECT DESCRIPTION

Background:

A patent, being a contract between the government and the person, is a complex document consisting of lengthy descriptions of scientific and legal criteria making them difficult to read and comprehend. Hundreds of such documents are received at patent offices every day and manual classification of such huge number of documents with highly specialized technical jargon makes the process slow, costly and also requires expert personnel for the process. For this purpose, automatic or semi-automatic patent classification codes were developed. In the process of automatic classification, codes are assigned by the computer program to the patent application on the basis of the content in the patent document. This makes the classification faster, systematic, accurate and reduces the error induces by the human process.

Generally, patents have a format which is a combination of structured and unstructured information.

Structured Data	Unstructured Data
Publication Number	Abstract
Classification Number	Title
Application Number	Summary
Filing Date	Background
Inventor/Assignee Information	Description
References cited	Claims

Also, a number of detailed figures present are a source of information, required for categorising the documents. The description section details the structure and function of invention giving specifics of domain to which the patent is related to. The claims section gives details of the legal protections given. In this project, we have considered the entire patent document for the purpose of classification into primary IPC sections.

The International Patent Classification (IPC) is agreed internationally where a patent can have several classification symbols but there is one which is the primary one. This is what is called the primary IPC mark. The documents are classified into primary IPC sections where each section symbol is denoted by a letter from letter A to letter H as follows:

‘A’ – “Human necessities”
‘B’ – “Performing operations, transporting”
‘C’ – “Chemistry, metallurgy”
‘D’ – “Textiles”
‘E’ – “Constructions”
‘F’ – “Mechanical engineering”
‘G’ – “Physics”
‘H’ – “Electricity”.

Automatic document classification can be categorized as: unsupervised, supervised and semi-supervised document classification. The classification by supervised model is enabled by an external mechanism like a human that supplies information for correct document classification which is opposite to the process of unsupervised classification where no such information is supplied by a mechanism. The two main factors that make document classification a challenging problem are (a) feature extraction (b) topic ambiguity. The process of extracting a set of features from the document in order to build a classification model that can classify the document accurately is called Feature Extraction. Extracting the right set of features is very critical as it determines the accuracy of document classification. The usage of standard textual similarity measures for classification of documents proves less accurate. Second, as documents have topics inter-relating different subjects, it causes ambiguity to categorize such documents. Also topics having different meaning may exist in the document which raises ambiguity in classification.

For this purpose, various algorithms and techniques are developed that can learn themselves by providing observed data. This is made possible as these algorithms construct stochastic models that enable them to make predictions and decisions. Some of such algorithms are Support Vector Machine, Naïve Bayes classifier, Neural Network, Decision Trees, etc. This area of subject is statistical machine learning. The classifier is trained in supervised document classification by providing a dataset of documents. The category of these documents is predicted and the confidence indicator is provided by the classifier. The quality of such predictions is affected by the data set that is provided for training purpose. The process of patent classification enables a quick search for the required document or any document that is related to the same subject or to keep a track of the similar technological trends in that area.

Aim:

The goal of this task is to automatically classify patents into principal International Patent Classification sections i.e., the first character (A to H) of the primary IPC sections.

Significance:

With the growing number and diversity of inventions, patent classifications are the need of the hour. Automatic text classification is a necessary invention that can classify the text into categories using statistical techniques that not only categorize automatically but also with speed and efficiency. It allows for more convenient and systematic organisation of documents that can be easily referred to whenever necessary. This makes it feasible to search quickly for documents about earlier documents that are similar to or are related to the invention for which a patent is applied for. It also facilitates to track the technological trends in a particular area in patent documents.

The automatic classification leads to more harmonized results than a human based process. It also saves time, cost and eliminated error that might get induced in the latter. Automatic Patent Classification is useful for other purposes such as economic intelligence, technological watch, etc.

Research Question:

- How to classify the patent documents automatically by statistical techniques?
- How Exploratory Data Analysis helps to summarize the characteristics of the dataset?
- Which machine learning algorithm provides better performance and accuracy for text classification?

Expected Outcomes:

In this project, a tool is designed which can predict the unlabelled patent documents into various categories of primary IPC sections (A to H) in a more accurate and scalable way possible by deploying an appropriate machine-learning algorithm. For the purpose of training the classifier, approximately 4000 labelled documents are used which can be visualised categorically in a chart. After the classifier is trained, 1000 unlabelled documents are considered which are to be predicted automatically into the primary IPC sections. A confusion matrix is used here to measure the performance of the machine learning algorithm. It summarizes the prediction results of the classification and also gives insight not only into errors made by the classifier but also the type of errors being made. This gives an edge in predicting the output. This categorization is visualized into a chart.

SECTION 2: LITERATURE REVIEW

Text document classification is an emerging field in the research of text mining. Initially, [Chakrabarti et al (1997, 1998)] had developed a Bayesian hierarchical system into three levels of subclasses. These authors pointed out that by using the known classifications of patents, further classification can be improved. [(Larkey 1998)] had designed a tool based on K-Nearest Neighbor approach for US patent codes in which he included phrases for indexing purpose which has improved the precision of the system in patent searching rather than categorization. The precision was focussed on by other authors. [(Kohnen et al, 2000)] had classified the patent into 21 categories with a precision of 60.6%.

Later, a number of approaches were developed such as support vector machines (SVM) [(Soumen Chakrabarti, 2003)], K-nearest neighbor (KNN) [(Eui-Hong (Sam), 2001)], Naive Bayes classification [(Andrew McCallum, n.d.)], Decision tree (DT) [(Stalzberg, 1993)], Neural Network (NN) [(S, 2004)], and maximum entropy [(Kamal Nigam, n.d.)].

Most researchers search online for already existing papers in their interest area. Based on search criteria the results are narrowed. [(Neethu, M., Rajasree), 2013] proposed a classification systems based on natural language processing techniques and k-means clustering algorithm for paper classification. [(Jain, T. I., Nemade, D.), 2010] describe a model that determines whether a sentence is polar or neutral and if polar, it is determined whether positive or negative. Thus this model uses sentiment for text classification. A three level strategy is followed by [(Esmin, A., Roberto L. De Oliveira Jr.), 2012] for determining the sentiment of twitter data.

A generic strategy for automatic text classification was presented by Dalal and Zaveri [(Mita K. Dalal, 2011)], which includes phases such as pre-processing, feature selection, using semantic or statistical techniques, and selecting the machine learning techniques such as Decision tree, SVM, Naive bayes, hybrid techniques). The key issues involved in text classification like handling a number of features, working with text that is unstructured, figuring out missing metadata, and choosing a machine learning technique that is suitable for training a text classifier were also discussed.

[(Porter, M.F., 1980)] Documents generally contain strings of characters which need to be transformed into a format that can be used for training the learning algorithms and for classification. The information retrieval research says that word stems are good to be used for representation to learn the algorithm which is derived by removing the case forms. Here, the algorithm replaces the words having the same stem with their stem-word [(Hastie, T., R. Tibshirani), 2009]. This reduces unrealistic feature variations, increases frequency count for the required feature. This process is used in this project.

One of the features extracted for the purpose of classification is the word frequencies [(Koster, Seutter, & Beney, 2003)]. Here, the documents are taken in the vector form which has word frequencies of the documents considered. One of such approach is “Bag-Of-Words” (BOW) where a frequency count of the words in the document is obtained (Jurafsky & Martin, 2014). The advantages of such a approach is that it is simple, fast and inexpensive. But on the negative side BOW approach retains less linguistic information as compared to other parsing approaches that retain more of such information (Lewis, 1998). Also, it is essential that a value is maintained for every vector which leads to sparsity problem where it is found that every vector has a high number of zeroes in their frequencies.

[(Tong, S. and D. Koller), 2001] have represented the document as a stemmed, TD-IDF vector. The irrelevant features were ignored by a making a list of common words. Due to the significance of these methods, these are implemented in this project.

[(Aurangzeb Khan et al.), 2010] reviewed some approaches of machine learning and strategies to classify text. The pre-processing phase is carried out by feature extraction and feature selection processes. The stemming, removal of stop words and tokenization is done during feature extraction. Later, using Term Frequency-Inverse Document Frequency (TF-IDF) approach which takes into account the relevant features between words and documents, vector space is constructed. These prove to be good performing techniques as per indications of information gain and chi square.

In the work of [(C. J. Fall), 2003], four classifiers are used for evaluation: K-Nearest Neighbour, Support Vector Machine (SVM) and Naïve Bayes (NB). Here, the pre-processing phase was carried out by using removal of stop words and stemming using Potter algorithm and terms of information gain. It is observed that Naïve Bayes and Support Vector Machine brought the best results at class level and at the subclass level, Support Vector Machine has outperformed others.

Tilve and Jain [(Amey K. Shet Tilve, 2017)] used three text classification algorithms (Naive Bayes, SVM for text classification, and the new implemented Use of Stanford Tagger for text classification) for text classification on two different datasets. On comparing with the above classification strategies, Naïve Bayes, being simple is good as a text classification model. Gogoi and Sarma [(Moromi Gogoi, 2015)] has highlighted the performance of using Naive bayes technique in classification of documents.

OWAKE system is the automated patent categorization that is being used in production environment to classify patent applications in various categories of Japanese Intellectual Property Cooperation Center (IPCC). It had been designed to handle the Japanese patent codes and can achieve precision of 90% in classifying into 38 categories. It is based on Rocchio-like algorithm that does classification roughly and it is followed by K-Nearest Neighbour for refinement in categorization. It uses full length patent documents and based on Japanese dictionary extracts the keywords (Kakimoto, 2003).

According to [(Kim, H., P. Howland, H. Park), 2005], SVMs are found to be the best among the classification strategies for text categorization purpose. For the experiment, they have considered a MEDLINE data which has 5 categories. In this each class consists of 500 documents and 1250 documents are considered for training and testing purposes. The classification is performed using SVM and K-NN and it is found that SVM has higher accuracy and also has reduction in space and time. It is known that in SVM's, the computational capacity and learning complexity is independent of feature space and this has been to its advantage in dealing with distinct number of terms in text classification. This also present strategies to reduce the dimension of text vectors to lessen the time and space complexity. Also, it presents decision functions which helps in dealing with the issue in which a document belongs to multiple classes.

Ian Christopher and Sydney Lin [(Christopher, 2011)] have classified the patents under hierarchical IPC system. They have done feature extraction using NLTK and employed algorithms like Logistic Repression, Linear SVC and Neural Networks.

John W. Hall [(HALL, 2017)] has classified the documents using various machine learning methods - Naïve Bayes, k-Means Clustering, Principal Component Analysis, Decision Trees, SVM, Logistic Regression and Artificial Neural Networks. It is observed that Decision trees got comparatively high F1 score than other algorithms.

In this paper, a classification model has been built on and evaluated according to a small dataset of categories and 4000 documents and 1000 documents for training and testing respectively. The result was validated by employing recall, statistical measures of precision, and their combination F- measure. Results showed that SGD is a good classifier.

SECTION 3: METHODOLOGY AND PLAN

The input data taken for training is explored such as number of documents that are categorised into different groups. The process of text classification involves various phases: Pre-processing phase, training phase, testing phase and Usage phase. Pre-processing of all the labelled and unlabeled documents is done, followed by the training phase in which the classifier is constructed from the labelled training prepared instances. Then, the testing that is responsible for testing the model by testing the prepared samples whose class labels are known but not used for training model. Finally, usage phase, since the model is prepared to be used for classification of new prepared data whose class label is unknown. The sample patent documents are collected from Kaggle. They are classified into labelled documents and unlabeled documents.

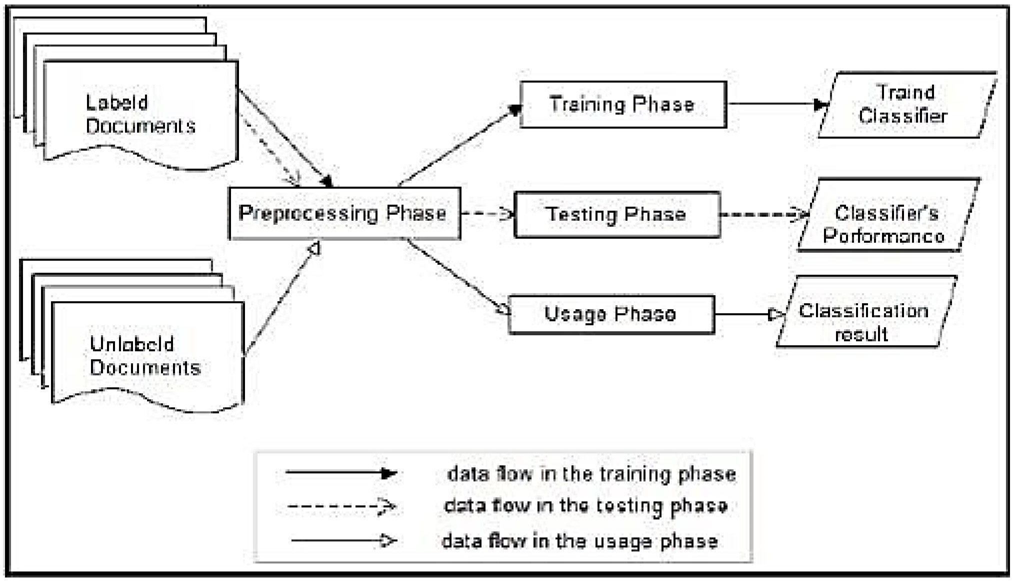


Figure 1: Model Architecture

Pre-Processing Phase: This phase is applied on the input documents - on the labelled data for training and testing phase or on the unlabeled for the usage phase. This phase gives the text documents in a word format. The output documents are prepared for next phases. Commonly the steps taken and are implemented in this project are:

- **Tokenization:** This step is implemented in this project for the purpose of conversion of the entire document into a list of token by treating the document as a string.
- **Removing stop words:** Here the stop words such as ‘a’, ’an’, ‘the’, ‘is’ etc. which are frequently used are insignificant and removed in this project.
- **Stemming word:** This process is implemented as it converts various word forms into a similar canonical forms. This step involves conflating tokens into their root form like converting ‘connection’ to its root form ‘connect’, ‘computing’ to its root form ‘compute’. The resultant documents are sent to training phase.

Training Phase: The documents prepared in the prepared in the Pre-processing phase, is used as input to this phase. Here, the documents are trained with an algorithm. The output of this phase is a trained classifier which can be tested and classified.

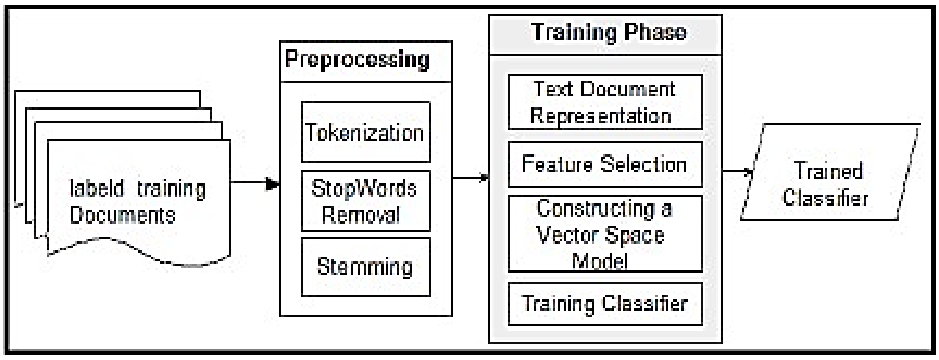


Figure 2: Training Phase

- **Text document representation**: It is the process of presenting the words and their number of occurrences in each document. There are two main approaches for this process - Bag of words (BOW) and Vector Space.

BOW - Each word is given a numerical value and indicated as a separate variable.

Vector Space – Here the text documents as vectors of identifiers.

Our model uses the BOW for representing the text documents using the Term Frequency-Inverse Document Frequency (TF-IDF).

- **Feature selection:** This process reduces the dimensionality of the feature set by removing the irrelevant features. This improves the efficiency of classification accuracy as well as reduces the computational requirements. TF-IDF and Stemming is used for this purpose in this project.
- **Training classifier:** The role of this phase is to build a classifier or generate model by training, using predefined documents that will be used to classify unlabeled documents. SGD Classifier is used for this purpose in this project.

Testing Phase: This phase is responsible for testing the performance of the trained classifier and evaluating its capability for the usage. The main inputs of this phase are the trained classifier and classified testing documents. They can be divided into: Unlabelled documents that are prepared during the pre-processing step and their associated class labels that can be used as input to the evaluation process.

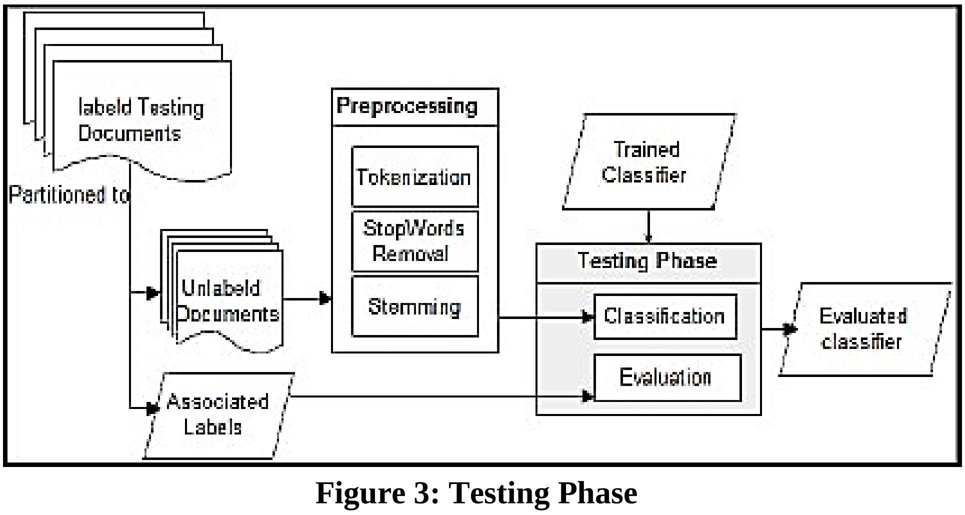


Figure 3: Testing Phase

- **Classification:** It is responsible for assigning an unlabelled document to the correct class of that document.

- **Evaluation:** In this step, a confusion matrix also called an error matrix which is a summary of predictions obtained on documents, is constructed. In construction matrix, the correct and incorrect categorization is summarized by assigning count values. In this manner the matrix shows the areas where the classification model is confused about classification i.e., it gets confused about the category in which the document should be put into. Thus, the matrix gives us details of the errors and the type of errors being committed by the classifier. The construction of the matrix describes the performance of the classification model on a set of data which is already predicted. The input of this step is the predicted class labels of the testing documents and the actual associated class labels. The performance of a classifier is evaluated according to the results of comparing the predicted class labels with the actual labels.

Usage Phase: The classifier in this phase is successfully trained, tested and evaluated and ready for classification of new data whose sections are unknown.

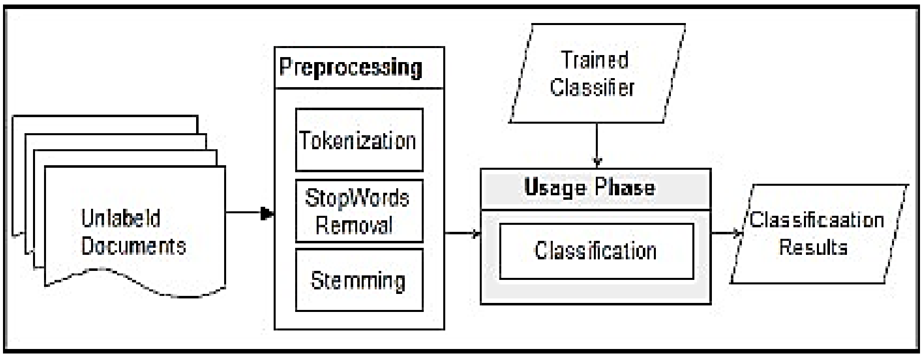


Figure 4: Usage Phase

TOOLS EMPLOYED:

PANDAS: It is an open source library that reads and analyses the data in the project. It also provides tools for data analysis, easy-to-use data structure and gives high-performance.

NLTK: It is a platform used to build Python programs that work with data of human language. It also gives easier-to-use interfaces like WordNet and also provides libraries for text processing for the purpose of classification, tokenization and stemming.

SCIKIT-LEARN: This is a library that is most useful for the purpose of machine learning in Python language and is also used for building classification models in this project.

MATPLOT: It is a plotting library that gives quality figures in a variety of formats and environments across various platforms. It is used for data-visualization purpose in this project.

TASK PLAN:

S.No.	TASK	REQUIRED TIME	STATUS
1	Input the required data from the data source	Week - 1	Completed
2	Feature extraction or pre-processing	Week – 2 to 4	Completed
3	Importing classification algorithms	Week – 5 to 7	Completed
4	Predicting the output and visualizing the output	Week - 7 to 9	Completed
5	Evaluation and Documentation of experiment	Week - 10 to 12	Completed

Week 1: Using PANDAS tool, data is taken from Kaggle in Jupyter notebook.

Week 2-4: By using NLTK tool, pre-processing is done by stemming, tokenization, removing stop words and TF-IDF vectorization.

Week 5-7: By using SKLEARN, the machine learning algorithms are imported and the algorithms are trained using labelled documents.

Week 7-9: With the trained algorithms, the unlabelled documents are predicted.

Week 10-12: By using confusion matrix available in SKLEARN, the classification output is evaluated. The entire experiment is documented.

SECTION 4: RESULTS AND EVALUATION

For the purpose of classification, 4972 documents are considered in this project. Of these, 3972 documents are labelled and are used for training the classification model and the rest are used for testing the model. The below graph shows the training data of the documents in each category.

IPC Sections	Number of Documents
A	1303
G	781
C	637
B	442
H	330
E	287
F	178
D	14

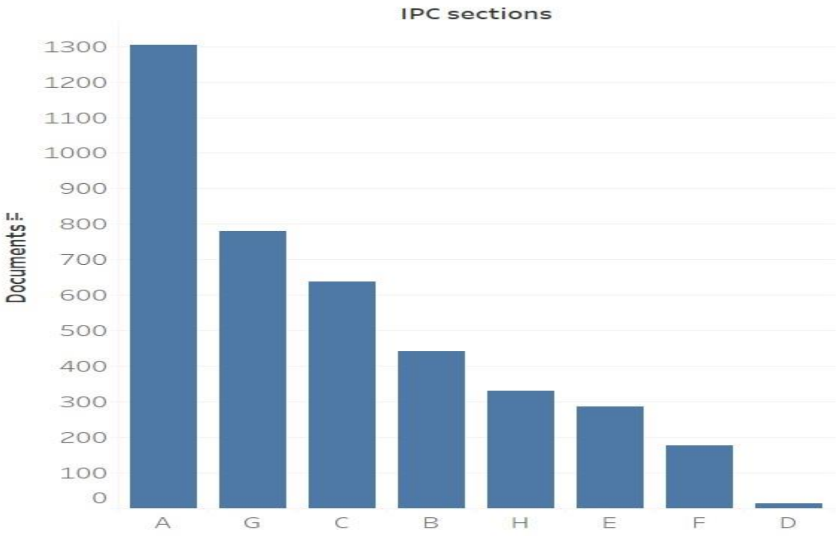


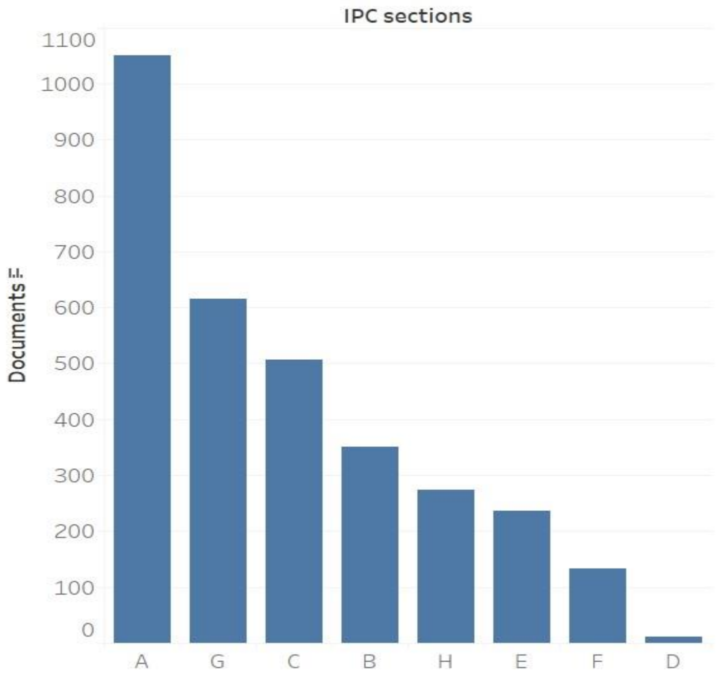
Figure 5: Data of training documents in each category

After pre-processing phase, the data is trained with various machine learning algorithms. All the algorithms are trained using the same data. In this project to find a better classification model, Random Forest classifier, Decision Tree classifier, Naive Bayes classifier and SGD classifier are considered. All the algorithms are subjected to parameter tuning. The results shown in below chart are those obtained after fine tuning of parameters. It is found that SGD classifier gives better accuracy of classifying the documents when compared with others.

Classifier	F1_Score
Decision Tree	**0.46**
Naïve Bayes	**0.51**
Random Forest	**0.58**
SGD	**0.70**
k-Nearest Neighbor	**0.65**
Gradient Boosting Classifier	**0.57**

As SGD classifier gives better accuracy and to evaluate the performance of the classifier model of SGD classifier, a confusion matrix is to be built. A confusion matrix is used here to measure the performance of the machine learning algorithm. It summarizes the prediction results of the classification and also gives insight not only into errors made by the classifier but also the type of errors being made. This gives an edge in predicting the output.

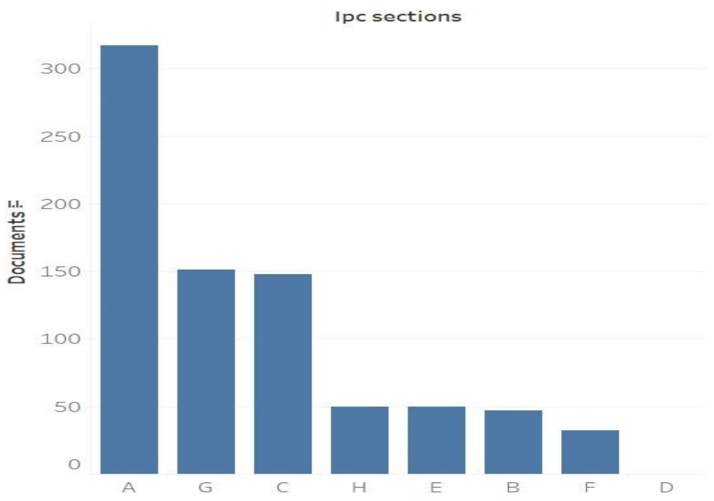
The training data (3972 documents) is split into training data (80%) and test data (20%). The training data in this case is visualised as the below graph:



IPC Sections	Number of Documents
A	1050
G	615
C	506
B	351
H	273
E	237
F	133
D	12

Figure 6: Data of training documents (80%) using SGD classifier

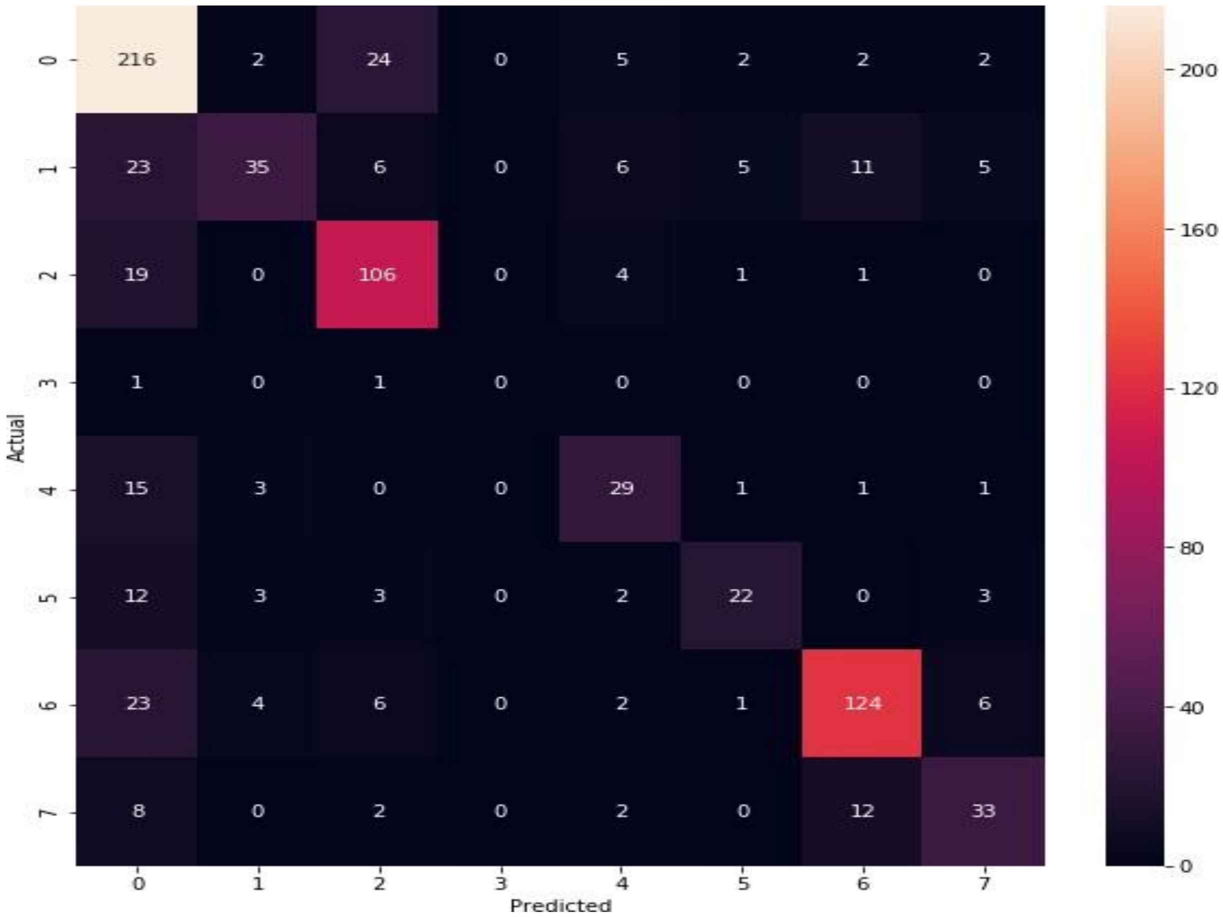
Here, after pre-processing phase, SGD classifier is used for training the data. The F1_score obtained as a result: 0.69. Now, using the classifier the test data (20%) is to be predicted. The documents categorised are visualised in the below graph:



IPC Section	Number of Documents
A	317
G	151
C	148
B	47
H	50
E	50
F	32
D	0

Figure 7: Data of predicted documents (20%) using SGD classifier

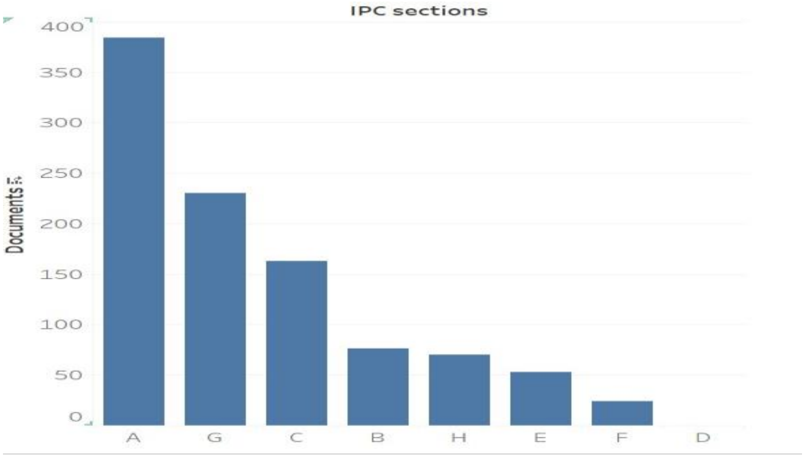
Now, the confusion matrix can be built using the actual data and predicted data.



Actual \ Predicted	0	1	2	3	4	5	6	7
0	216	2	24	0	5	2	2	2
1	23	35	6	0	6	5	11	5
2	19	0	106	0	4	1	1	0
3	1	0	1	0	0	0	0	0
4	15	3	0	0	29	1	1	1
5	12	3	3	0	2	22	0	3
6	23	4	6	0	2	1	124	6
7	8	0	2	0	2	0	12	33

Figure 8: Confusion Matrix

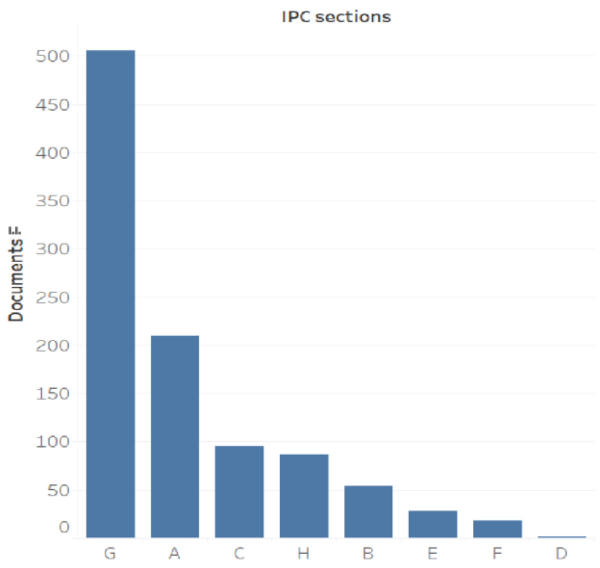
The predicted data is shown on x-axis while the actual data on y-axis. The representation of A category corresponds to 0, B to 1, C to 2 and so on. For the A category, it is observed that while the actual data is 216 documents, the predicted output also categorizes 216 documents into A category. However, the predicted output categorizes the 23 documents of B category into A category which is a mismatch and hence the color of that cell is a bit lighter than black. The colour is indicated based on the number of documents as shown in the legend in the right. Here, black indicates zero documents in that category whereas white indicates 216 documents. In this way the entire confusion matrix can be comprehended. Now using the SGD Classifier, the unlabeled documents are to be predicted.



IPC Section	Number of Documents
A	384
G	230
C	163
B	76
H	70
E	53
F	24
D	0

Figure 9: Data of predicted unlabelled documents

The classification of the documents by SGD Classifier is successful and the number of documents categorized are as shown above. In order to test the tool further, a different set of 1000 documents are considered.



IPC Section	Number of Documents
G	506
A	210
C	95
H	87
B	54
E	28
F	18
D	2

Figure 10: Predicted data of 1000 documents for testing the tool

CONCLUSION:

In this paper, we present a model for patent classification that depicts the stream of phases through building automatic text document classifier and presenting the relationship between them. Evaluation of the proposed model was performed on patents dataset. The experiment results confirmed that firstly, superiority of using the SGD classifier with TFIDF than decision tree and random forest classifiers as an approach for patent document classification. The proposed model demonstrates that the proposed choice of statistical techniques in the model gave better results, improved the classification process performance, and confirmed our concept of the compatibility between the selected techniques of various phases. Finally, using the SGD classifier model with better performance, the unlabelled patent documents are classified into 8 principal International Patent Classification sections.

REFERENCES

1. Amey K. Shet Tilve, S. N. J., 2017. Text Classification Using Naïve Bayes, Vsm And Pos Tagger. International Journal of Ethics in Engineering & Management Education, 4(1), p. 6.
2. Andrew McCallum, K. N., n.d. A Comparison of Event Models for Naive Bayes Text Classi. [Online]
3. Available at: http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf [Accessed 25 September 2018].
4. Anne, C., 2017. Advanced Text Analytics and Machine Learning. [Online] Available at: https://scholarworks.uno.edu/cgi/viewcontent.cgi?article=3466&context=td [Accessed 14 August 2018].
5. Christopher, I., 2011. Automated Patent Classification. [Online]
6. Available at: http://cs229.stanford.edu/proj2011/ChristopherLinSpieckermann Automated PatentClassification.pdf [Accessed 13 August 2018].
7. Eui-Hong (Sam), H. G. K. V. K., 2001. Text Categorization Using Weight Adjusted.
8. Guyot, K. B. a. J., 2011. Automated Patent Classification. The Information Retrieval.
9. HALL, J. W., 2017. Examination Of Machine Learning Methods For Multi-Label Classification Of Intellectual Property Documents. Illinois: s.n.
10. Kamal Nigam, J. L., n.d. Using maximum entropy for text classification. [Online] Available at: https://www.cc.gatech.edu/~isbell/reading/papers/maxenttext.pdf [Accessed 15 August 2018].
11. Khyati S. Kava, P. N. P. D., n.d. A Survey On Text Categorization Of Indian And NonIndian Languages Using Supervised Learning Techniques. [Online] Available at: https://www.researchgate.net/profile/Nikita_Desai3/publication/280229005_A_survey_on _text_categorization_of_indian_and_nonindian_languages_using_supervised_learning_te chniques/links/55c1b69f08ae9289a09d19fb/A-survey-on-text-categorization-of-indian and-n [Accessed 4 August 2018].
12. Mattyws F. Grawe, C. A. M. A. G. B., 2017. Automated Patent Classification Using Word. s.l., IEEE International Conference on Machine Learning and Applications.
13. Mita K. Dalal, M. A. Z., 2011. Automatic Text Classification: A Technical Review.
14. International Journal of Computer Applications, 28(2), p. 4.
15. Moromi Gogoi, S. K. S., 2015. Document Classification of Assamese Text Using Naïve Bayes Approach. International Journal of Computer Trends and Technology, 30(4), p. 5.
16. Reza, M., 2017. Network Traffic Classification using Machine Learning Techniques over Software Defined Networks. International Journal of Advanced Computer Science and Applications.
17. Soumen Chakrabarti, S. R. M. V. S., 2003. Fast and accurate text classification via multiple linear discriminant projections. The VLDB Journal, p. 16.
18. Stalzberg, S. L., 1993. Programs for Machine Learning. Netherlands: Kluwer Academic.
19. S, W., 2004. Neural network agents for learning semantic, s.l.: s.n.
20. C. J. Fall, A. T¨orcsv´ari, K. Benzineb, and G. Karetka, “Automated categorization in the international patent classification,” ACM SIGIR Forum, vol. 37, no. 1, pp. 10–25, apr 2003. [Online]. Available: ttp://portal.acm.org/citation.cfm?doid=945546.945547.

21. D. Tikk, G. Bir´o, and A. T¨orcsv´ari, “A hierarchical online classifier for patent categorization A hierarchical online classifier for patent categorization,” Emerging Technologies of Text Mining: Techniques and Applications, vol. 1, no. 1, pp. 244–267, 2008.