Journal of Advances in Science and Technology
Vol. 18, Issue No. 2, September–2021, ISSN 2230-9659

An Analysis the Document Classification by Using Learning Techniques

Ankur Pandey[1]*, Dr. Anoop Kumar Chaturvedi[2]

[1] Research Scholar, LNCT University

[2] Guide, LNCT University

Abstract - Text classification intends to provide high quality textual representation accessed from the digital forms of document available online and build high quality classifiers. Current research explores the Machine learning classifiers for text classification. In order to extract models, classification algorithms are used to describe important data classes. Classification of documents can be supervised, semi-supervised or unsupervised. Using text classification methods such as Random forest (RF), Xgboost, Naive Bayes Classifier, Logistic regression. Neural network based models are widely used and outperforms other models but they take more time for training, thereby limiting their usage on large datasets. The precision is defined as the percentage of properly retrieved documents that are related to the query. The process of text classification begins with identifying ideal features and selection of machine learning classifiers. All evaluation metrics show that the proposed work improves retrieval performance.

Keywords - Text Classification, Document Classification, Machine Learning, Evaluation Metrics

----------X----------

INTRODUCTION

NLP has several applications in web search, retrieval of information, ranking and classification of documents where text classification is an important task. Text classification intends to provide high quality textual representation accessed from the digital forms of document available online and build high quality classifiers [Xu, JS 2007]. The phases involved in text classification are database collection, preprocessing of the data, reduction in dimensionality of the dataset and implementation of the classifier. Current research explores the Machine learning classifiers for text classification. Some of the effective classifiers are Naive Bayesian, support vector machine, k-nearest neighbor classification, neural network and so on. Neural network based models are widely used and outperforms other models but they take more time for training, thereby limiting their usage on large datasets. A neural network obtains state of the art performance when they are trained with the suitable ideal features and scales to a larger corpus [OzgurLevent 2010].

Document Classification (DC) stands as the process of analyzing a group of documents and labeling each one of them with an appropriate category as per its relevance towards one of the pre-defined collection of categories (Joorabchi& Mahdi 2011). The DC is rife with potential for several modern document-centric applications, like Document Summarization, essay scoring, organizing documents for query-based information dissemination, email management and topic-specific search engines. DC is usually a 2-step process. First, the text contained in a document is analyzed and a compact set of features which characterize the document are generated. Next, centered on their extracted features, the documents are classified to their respective categories by employing a suitable ML technique.

DOCUMENTS CLASSIFICATION

In the field of information extraction & retrieval, document classification is regarded as a classic problem. It is crucial in many contexts, as it facilitates the management of text and enormous amounts of unstructured data. There are two main approaches that can be taken to classify documents: manually or automatically. As was previously said, in manual document classification, people interpret the meaning of the text and other elements in order to establish the relationships between concepts & categorize documents. Automatic document categorization, on the other hand, employs sophisticated methods like ML & Deep Learning to categorize documents without human intervention. When compared to human classification, this method is quicker, more scalable, more accurate, and cheaper. Let's get a handle on the various papers there are to classify automatically before we get into the various methods:

1. **Structured Documents:** The information is neatly organized in structured documents or other set formats. There are not many

variations in the fonts or the numbers. These formats, or templates, are completely unchanging. Since structured documents tend to follow a predictable pattern, it's not hard to create an automated solution on top of them.

2. **Unstructured Documents:** Letters, orders, contracts, & bills of lading are all examples of openly formatted documents. Unstructured papers make it difficult to find specific information because their coordinates are all over the place. Sometimes the tables in these papers don't even have borders, making it harder for computers to pinpoint their precise location. Building clever algorithms for such materials typically involves extensive use of NLP and NER techniques.
3. **Semi Structured Documents:** In most cases, semi-structured documents are a hybrid form that combines elements of both structured & unstructured formats.

A variety of methods exist for classifying documents.

Supervised Learning: This technique uses examples with both inputs & classes or results that can be expected from them to teach the system how to improve its performance. The program learns on data that has been annotated by humans. Once the classifier has been trained, it can make predictions about categories along with a confidence indication, even for data/document types it has never encountered before.

Unsupervised Learning: Using this method, publications with similar content are automatically grouped into distinct categories. Some examples of criteria that could be used for this sorting are the use of a specific design theme, the use of specific words in the font, the presence of specific tags, etc. If proper guidelines are created and fine-tuned, these algorithms can attain more accuracy.

Rule-based: One of the most conventional approaches to document categorization is the rule-based strategy, which involves taking use of a computer system's ability to interpret natural language and establishing grammatical rules to teach the computer how to classify documents like a human would. With this approach, rather of relying simply on statistics or mathematics, performance can be improved on a regular basis. Increased precision is typically linked with this approach. However, developing a cutting-edge rule-based model is laborious and restricts scalability.

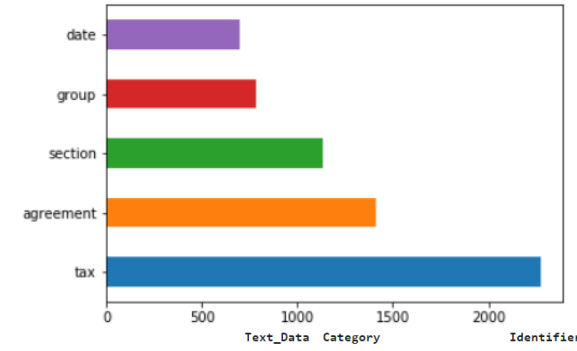
STEPS

1. Import library of python

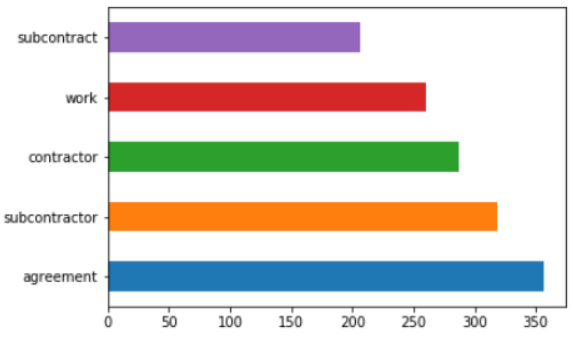
```
# Import the library of python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

```

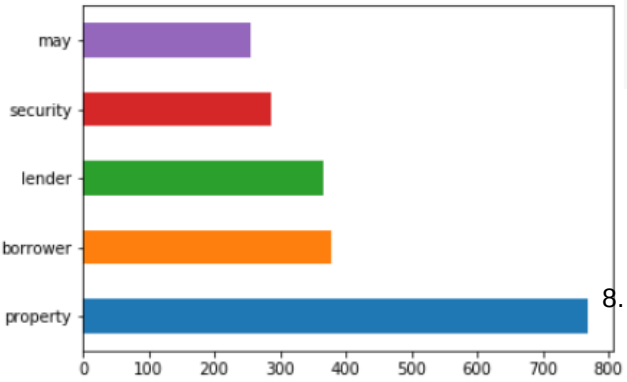
2. The raw data contains different file formats. Data needs to be extracted from them. Here doc files have been converted to PDF format first before extraction. I had issues while installing textract library which extracts test from the doc files directly, so I converted them to PDF & utilizedpdfminer to extract texts. Texts are extracted on folder basis because the extracted texts need to be labeled.
3. Extract text data from pdf and other doc format
4. Extract text files and label the data
5. In this data we have Agreements, Deeds, Human Resource, Images, Taxes, and valuations
6. Perform EDA



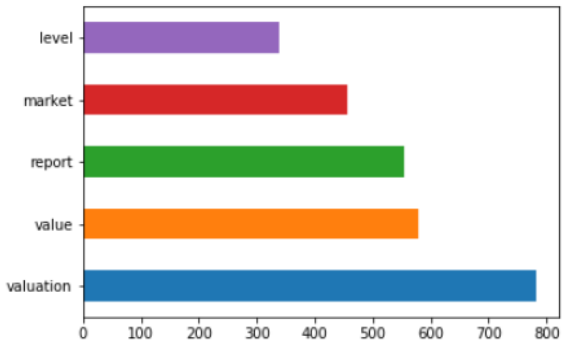
	Text_Data	Category	Identifiers
0	tax indemnity agreement tax indemnity agreemen...	Taxes	tax,agreement,section,group,date
1	factor ii acquisition price series c share iss...	Taxes	tax,agreement,section,group,date
2	earnout dilution factor means respect earnout ...	Taxes	tax,agreement,section,group,date
3	turn appoint third appraiser determine fair v...	Taxes	tax,agreement,section,group,date
4	way diminution value notwithstanding anything ...	Taxes	tax,agreement,section,group,date



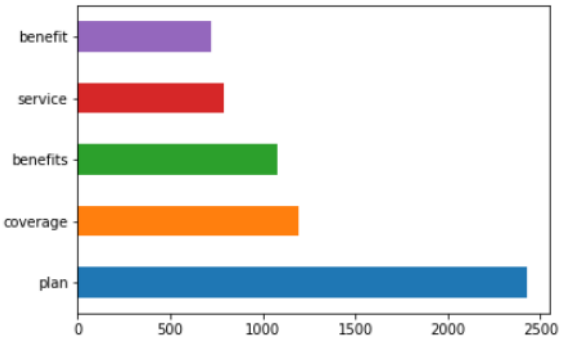
	Text_Data	Category	Identifiers
0	agreement xxw amendment agreement among united...	Agreement	agreement,subcontractor,contractor,work,subcon...
1	act september stat designated colorado river b...	Agreement	agreement,subcontractor,contractor,work,subcon...
2	section party provides additional funding part...	Agreement	agreement,subcontractor,contractor,work,subcon...
3	witness whereof parties hereto executed amendm...	Agreement	agreement,subcontractor,contractor,work,subcon...
4	witness whereof parties hereto executed amendm...	Agreement	agreement,subcontractor,contractor,work,subcon...



	Text_Data	Category	Identifiers
0	sample deed trust deed trust definitions words...	Deeds	property,borrower,lender,security,beneficiary
1	cid adjustable rate rider cid condominium ride...	Deeds	property,borrower,lender,security,beneficiary
2	q successor interest borrower means party take...	Deeds	property,borrower,lender,security,beneficiary
3	check drawn upon institution whose deposits in...	Deeds	property,borrower,lender,security,beneficiary
4	b leasehold payments ground rents property c p...	Deeds	property,borrower,lender,security,beneficiary



	Text_Data	Category	Identifiers
0	valuation reportbusiness valuation trends stra...	Valuations	valuation,value,report,market,level
1	valuation report spring technical topicsthe im...	Valuations	valuation,value,report,market,level
2	spring valuation reportlitigation update dlom ...	Valuations	valuation,value,report,market,level
3	publication distributed understanding author p...	Valuations	valuation,value,report,market,level
4	november actuarialvaluationreportguaranteededu...	Valuations	valuation,value,report,market,level



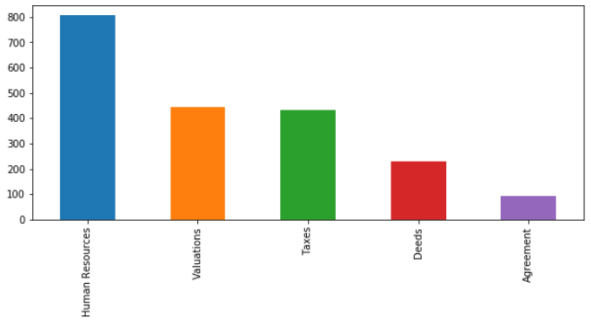
	Text_Data	Category	Identifiers
0	volunteer coordinators job description general...	Human Resources	plan,coverage,benefits,service,level
1	cid participate volunteer evaluations assigned...	Human Resources	plan,coverage,benefits,service,level
2	hiring company name texas ballet theater hirin...	Human Resources	plan,coverage,benefits,service,level
3	oversee community enrichment citydance outreac...	Human Resources	plan,coverage,benefits,service,level
4	dynamic energetic creative socially adept demo...	Human Resources	plan,coverage,benefits,service,level

7. Merging all data file s into single pandas dataframe

```
frames = [dftaxes, dfagreement, dfdeeds, dfvaluation, dfhr]
finalframe = pd.concat(frames,sort=False)
finalframe = finalframe[['Identifiers','Text_Data','Category']]
finalframe = finalframe.reset_index(drop=True)
finalframe[:5]
```

	Identifiers	Text_Data	Category
0	tax,agreement,section,group,date	tax indemnity agreement tax indemnity agreemen...	Taxes
1	tax,agreement,section,group,date	factor ii acquisition price series c share iss...	Taxes
2	tax,agreement,section,group,date	earnout dilution factor means respect earnout ...	Taxes
3	tax,agreement,section,group,date	turn appoint third appraiser determine fair v...	Taxes
4	tax,agreement,section,group,date	way diminution value notwithstanding anything ...	Taxes

8. Count plot for each doc type



9. Perform TFIDF conversion of text data.

TF-IDF = Term Frequency (TF) * Inverse Document Freq

Term Frequency

It counts how many times a certain words appears in a given text. How many times a word like "was," which is extremely common, seems in a document is strongly dependent on the document's length and the word's generality. Take two texts, one with 100 words and the other with 10,000; the former is likely to have more instances of the ubiquitous "was" than the latter. However, the lengthier document cannot be considered more significant than the shorter one. This is why we employ normalization on the frequency value, which involves dividing the frequency by the total number of words in the document.

Document Frequency

The significance of documents across the entire corpus is quantified in this manner. This is quite similar to TF; the only distinction is that DF counts how many times a term t appears in the entire document set N, whereas TF just counts how many times the term t appears in document d. The frequency with which a word seems in written work is denoted by its document frequency (DF). When counting the occurrences of a term in a document, we only count the first occurrence without regard to how many times the phrase really appears.

df(t) = occurrence of t in N documents

This is also normalized by divide by the total numbers of documents to ensure it remains within a range. If we want to know how informative a word is, then we need to calculate its DF, which is the opposite of what we want to

do. Because of this, we flip the DF.

Inverse Document Frequency

A term's informativeness, represented by t, can be quantified by its inverse document frequency (IDF). We should expect a very small IDF value for highly frequent terms like stop words (because these words appear in virtually all of the papers and the N/df statistic will assign a little value to this word). As a result, we can now get a relative weightage, which was our original goal.

idf(t) = N/df

it will be skipped over at question time. But in certain circumstances, we employ a set vocab and few terms of the vocab might be absent in the document, in such cases, the df will be 0. Given that we are unable to divide by zero, we will add one to the denominator in order to round off the value.

$$idf(t) = \log(N/(df + 1))$$

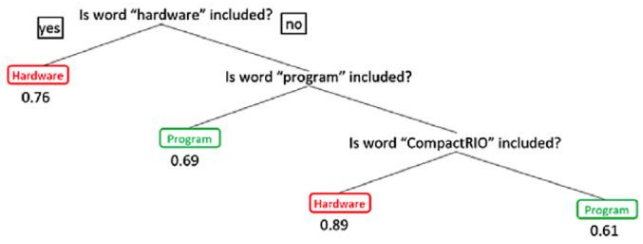
The TF-IDF score is calculated by multiplying the TF score by the IDF score. The TF-IDF can be modified in various ways, but for the time being, let's just stick with this elementary implementation.

$$tf\text{-}idf(t, d) = tf(t, d) * \log(N/(df + 1))$$

10. Divide data into train test split in the ratio of 70:30
11. Perform oversampling for better-balanced classes
12. Machine learning algorithms implementation

Random forest

As an ensemble learning method, random forests in machine learning are effective in resolving regression & classification issues. As a result, it is able to solve difficult problems by combining the strengths of various classifiers. Multiple decision trees are aggregated by bagging or bootstrapping to form a random forest, which is effectively an algorithm.The mean output of the decision trees is used to make predictions in a random forest text categorization model.



Text Classifiers in Machine Learning: Random Forest.

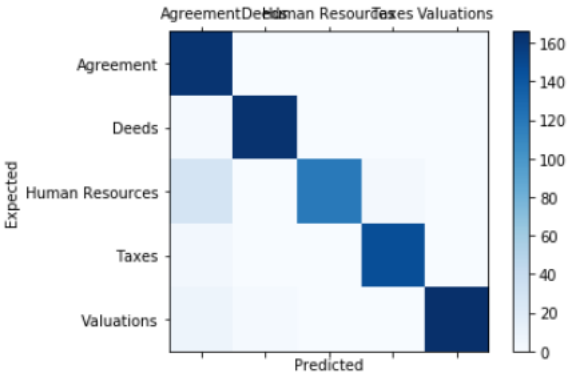
Now implementing the Random Forest using Python.

```
Accuracy: 0.9405204460966543
                precision    recall  f1-score   support

    Agreements       0.79      1.00      0.88       163
         Deeds       0.99      0.99      0.99       166
         Taxes       1.00      0.79      0.88       151
Human Resources      0.98      0.97      0.98       151
    Valuations       1.00      0.94      0.97       176

     micro avg       0.94      0.94      0.94       807
     macro avg       0.95      0.94      0.94       807
  weighted avg       0.95      0.94      0.94       807

Confusion matrix:
 [[163   0   0   0   0]
 [  2 164   0   0   0]
 [ 29   0 119   3   0]
 [  4   0   0 147   0]
 [  8   2   0   0 166]]
```



The effectiveness of a particular classification model (or "classifier") on test data for which the true values are known can be summarized in a table called a confusion matrix.

Both the X-axis (True labels) and Y-axis (Predicted labels) show document classes that we used for training. Numbers inside the cells represent the share of the testing dataset that falls into the left & bottom categories.

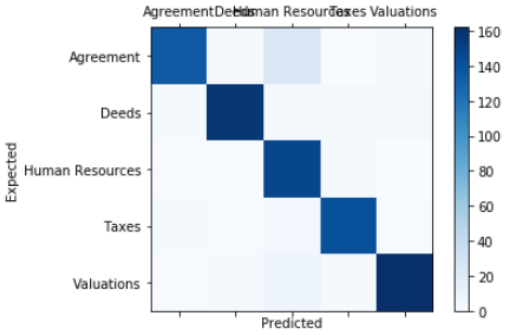
Naïve Bayes classifier

Accuracy: 0.9231722428748451

	precision	recall	f1-score	support
Agreements	0.96	0.83	0.89	163
Deeds	0.96	0.95	0.95	166
Taxes	0.80	0.98	0.88	151
Human Resources	0.96	0.94	0.95	151
Valuations	0.96	0.92	0.94	176
micro avg	0.92	0.92	0.92	807
macro avg	0.93	0.92	0.92	807
weighted avg	0.93	0.92	0.92	807

Confusion matrix:

```
[[136   2  23   0   2]
 [  2 157   2   2   3]
 [  1   0 148   2   0]
 [  2   1   5 142   1]
 [  1   3   8   2 162]]
```

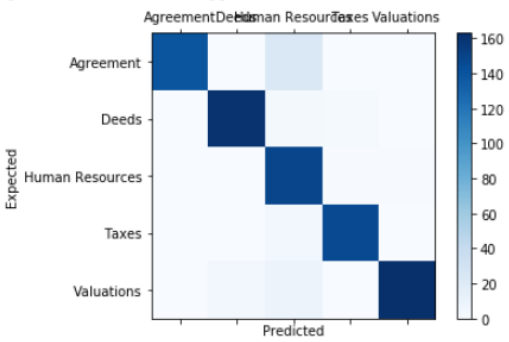


Accuracy: 0.9417596034696406

	precision	recall	f1-score	support
Agreements	1.00	0.86	0.92	163
Deeds	0.98	0.97	0.97	166
Taxes	0.79	0.99	0.88	151
Human Resources	0.98	0.97	0.98	151
Valuations	0.99	0.93	0.96	176
micro avg	0.94	0.94	0.94	807
macro avg	0.95	0.94	0.94	807
weighted avg	0.95	0.94	0.94	807

Confusion matrix:

```
[[140   0  23   0   0]
 [  0 161   3   2   0]
 [  0   0 149   1   1]
 [  0   0   4 147   0]
 [  0   4   9   0 163]]
```

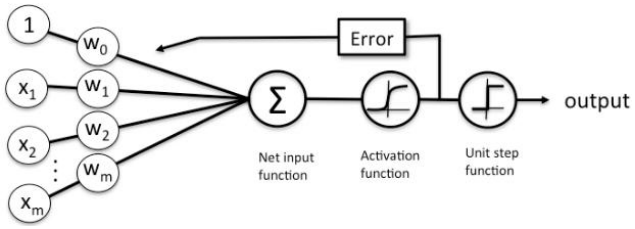


Xgboost

The tree-based model family also includes boosting models, which are another kind of ensemble model. Boosting is a group of ML algorithms that transforms weak learners into strong ones; it is aensemble meta-algorithm for mainly reduce bias & variation in supervised learning. Poor learners are characterized as classifiers whose correlation with the correct classification is weak (it can label examples better than random guessing)

Logistic regression

Logistic regression is a supervised learning approach typically used to manage binary "classification" tasks while having the word "regression" in its name. In spite of the inherent incompatibility between "regression" and "classification," the emphasis in logistic regression is on the word "logistic," which stands for the logistic function that carries out the classification in the method.



Schematic of a logistic regression classifier

accuracy 0.9385382059800664

	precision	recall	f1-score	support
Agreements	0.84	0.81	0.82	26
Deeds	0.91	0.94	0.92	71
Taxes	0.92	0.98	0.95	244
Human Resources	0.98	0.88	0.92	136
Valuations	0.98	0.94	0.96	125
micro avg	0.94	0.94	0.94	602
macro avg	0.92	0.91	0.92	602
weighted avg	0.94	0.94	0.94	602

Evaluating the performance

A popular technique for assessing a text classifier's performance is cross-validation. It operates by randomly creating example sets of similar length from the training dataset (for instance, four sets having 25% of the data). The leftover samples from each set (for instance, 75% of the samples) are used to train a text classifier. The classifiers then make predictions on their respective sets, & outcomes are contrasted with the tags that humans have annotated. This will reveal if a prediction was accurate (true positives & true negatives) or incorrect (false positives, false negatives).

These findings allow you to create performance indicators that are helpful for a rapid evaluation of classifier performance:

- Accuracy: the proportion of texts that were correctly tagged when they were categorised.
- Precision: the proportion of correctly classified instances out of all those predicted for a specific tag by the classifier.
- Recall: the proportion of cases the classifier correctly predicted for a specified tag out of all the examples it ought to have correctly predicted.
- The harmonic mean of recall & precision is the F1 score.

CONCLUSION

This work was implemented text-mining methods to resolve the problematic by automatically classifying text documents where one put unknown document content and found the class of the document as per the output of genetic algorithm by pattern presented in content. In order to filter the patterns from the document frequent words which cross threshold is considered as the patterns. This approach is very user friendly with security of whole content and less time consuming. Additionally, the proposed work has enhanced the work's accuracy. The provided dataset is utilized to verify the effectiveness of machine learning methods. The conventional ML algorithms in terms of accuracy, precision, recall, f1-score, & support, as determined through experimental evaluation.

REFERENCES

1. Abualigah, L. M., &Khader, A. T. (2017). Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. The Journal of Supercomputing, 73(11), 4773-4795.
2. Abualigah, L. M., Khader, A. T., &Hanandeh, E. S. (2018). A new feature selection method to improve the document clustering using particle swarm optimization algorithm. Journal of Computational Science, 25, 456-466.
3. Abualigah, L. M., Khader, A. T., Al-Betar, M. A., &Alomari, O. A. (2017). Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. Expert Systems with Applications, 84, 24-36.
4. Abualigah, L. M., Khader, A. T., Hanandeh, E. S., &Gandomi, A. H. (2017). A novel hybridization strategy for krill herd algorithm applied to clustering techniques. Applied Soft Computing, 60, 423-435.
5. Adrian Bilski 2011, 'A review of artificial intelligence algorithms in document classification', International Journal of Electronics and Telecommunications, vol. 57, no. 3, pp. 263-270.
6. Aggarwal, CC &Zhai, C 2012, 'A survey of text classification algorithms in mining text data', Springer, Boston, MA, pp. 163-222.
7. Aghdam, M. H., &Heidari, S. (2015). Feature selection using particle swarm optimization in text categorization. Journal of Artificial Intelligence and Soft Computing Research, 5.
8. Agnihotri, D., Verma, K., &Tripathi, P. (2017). Variable global feature selection scheme for automatic classification of text documents. Expert Systems with Applications, 81, 268-281.
9. Ahmad BasheerHassanat, Mohammad Ali Abbadi, GhadaAwadAltarawneh& Ahmad Ali Alhasanat 2014, 'Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach', arXiv preprint arXiv:1409.0919.
10. Ahmed H Aliwy&Esraa H Abdul Ameer 2017, 'Comparative study of five text classification algorithms with their improvements', International Journal of Applied Engineering Research, vol. 12, no. 14, pp. 4309-4319.
11. E Jadon, R Sharma et al. "Data Mining: Document Classification using Naive Bayes Classifier" International Journal of Computer Applications Volume 167 – No.6, June 2017.
12. Elvis Saravia, Carlos Argueta &Yi-Shin Chen 2016, 'Unsupervised graphbased pattern extraction for multilingual emotion classification', Social Network Analysis and Mining, vol. 6, no. 1, P. 92.
13. Eric P Xing, Rong Yan & Alexander G Hauptmann 2012, 'Mining associated text and images with dual-wing harmoniums', arXiv preprint arXiv:1207.1423.
14. Farman Alia, Kyung-Sup Kwaa,Yong-GiKimb," Opinion mining based on fuzzy domain ontology and Support Vector Machine: A proposal to automate online

review classification", Applied Soft Computing-2016.
15. Feldman, R., and J. Sanger. 2006. The text mining handbook: Advanced approaches in analysing unstructured data. Cambridge: Cambridge University Press.
16. Filippo Maria Bianchi, Simone Scardapane, AntonelloRizzi, Aurelio Uncini&AlirezaSadeghian 2016, 'Granular computing techniques for classification and semantic characterization of structured data', Cognitive Computation, vol. 8, no. 3, pp. 442-461.
17. Fnedman J. (1994). 'Flexible metric nearest neighbor classification', technical report 113, Stanford university statistics department, Stanford.
18. Francisco Charte, Antonio Rivera, María José del Jesus & Francisco Herrera 2012, 'Improving multi-label classifiers via label reduction with association rules', in International Conference on Hybrid Artificial Intelligence Systems. Springer, Berlin, Heidelberg, pp. 188-199.

Corresponding Author

Ankur Pandey*

Research Scholar, LNCT University