National College of Ireland

A Study of Different Pre-Processing Approaches of Text Categorization

MSc Research Project
Data Analytics

Sharan Prabhulinga Narke
x16110242

School of Computing
National College of Ireland

Supervisor: Dr. Simon Caton

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Sharan Prabhulinga Narke
Student ID:	x16110242
Programme:	Data Analytics
Year:	2016
Module:	MSc Research Project
Lecturer:	Dr. Simon Caton
Submission Due Date:	11/12/2017
Project Title:	A Study of Different Pre-Processing Approaches of Text Categorization
Word Count:	5566

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	10th December 2017

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Study of Different Pre-Processing Approaches of Text Categorization

Sharan Prabhulinga Narke
x16110242
MSc Research Project in Data Analytics

10th December 2017

Abstract

Text Pre-processing is a process of converting raw text data in to corpus(bag of words) which is further fed into different classifiers for text categorization. This paper presents the results of an experimental study of some text pre-processing techniques used against various classification algorithms.The main intent is to understand and discover best possible pre-processing technique to procure better classifier performance. In particular, text pre-processing techniques like Document Term Matrix(DTM), Term Document matrix(TDM) and Term Frequency-Inverse Document Frequency(TF-IDF) were used against 10 different classifiers on BBC News dataset. A comparative performance analysis of classifiers is conducted using evaluation metrics like Accuracy, Precision, Recall and F-score. The results indicate TF-IDF as better pre-processing method aiding better classifier performance than DTM and TDM.

1 Introduction

Text Analytics can be simply defined as the process of transforming unstructured text data into a systematic structured format which can be further be used to derive useful insights (Vandierendonck et al.; 2016). Text Analytics holds a valuable position in the field of data analytics. By converting text data from unstructured to structured format helps in the terms of Sentimental analysis, categorization, Spam e-mail filtration.

Lately, we have seen an outburst of text content generated on web-pages, social-networking websites and e-mails. The pace at which the content gets generated is enormous. This urges us to find out the ways by which the text analytics can cope up with such a huge scale without compromising the quality of analysis. Availability of the data can vary in terms of its structure so does the processing. The critical step that involved in text analytics is pre-processing of data which is be further used for various text analysis procedures.Analyzing text data (Big data) and deriving insight out is made possible by use of Machine learning approaches which have turned out to be effective on all grounds and made possible to achieve results in a more effective manner. Feature selection methodology is most effective method which has worked wonders in field of text analytics.

However, this methodology might get saturated in its effectiveness with existing supporting pre-processing procedures and it forces to look out ways of different pre-processing

or combinations to render maximum performance from Feature selection methodology. The fact why analyzing text is important for organization is because they have 80percent[1]of unstructured data in the form of text which is unused, For which when used can help to get deep insights and hence help in taking effective business decisions. Below figure shows how importance of Text analytics has increased over the years.

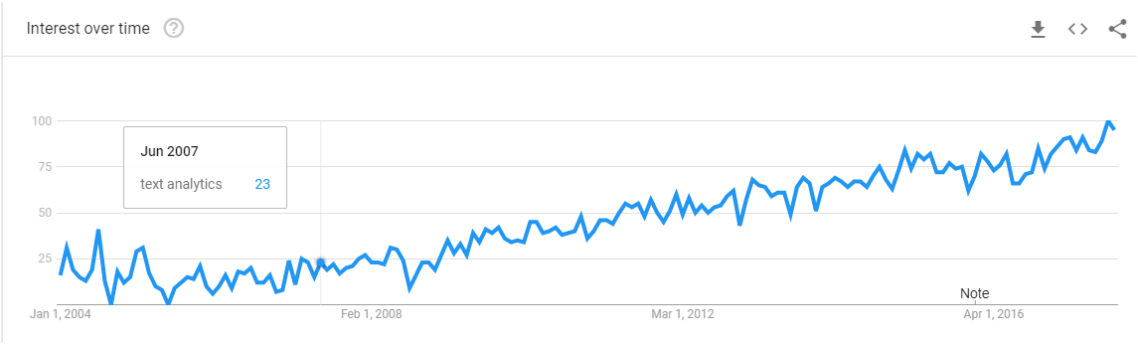

Figure 1: Text Analytics trend from 2004-Present

This trend has been illustrated by (Ittoo et al.; 2016).Industries have understood the importance of text data its potential value and how this data can be used to formulate safety reports, company specific documents etc. Huge organizations like Netflix, Bank of England etc. are using this potential data to extract analytics from social media, blogs in order to understand the sentiments among their customers and improvise accordingly. Figure 1 shows how the importance of Text analytics has increased over the years[2]

The main intent of this research is to explore and understand the attributes or contingent factors result in a more effective means of pre-processing text data for categorization purposes. Since major part of Text classification involves pre-processing of the Text data(Documents) which can be further fed into the Machine learning models for analyzing. The results from machine learning models is then compared with the results obtained from Ant colony optimization as it exhibited very impressive results while performing classification on a multi-labelled dataset by creating different classification rule in a research carried out by (Martens et al.; 2007) to let know which method would be more suitable to deliver high quality results and what would be depending factors that drive that particular result. The whole research can be divided in to following tasks:

Research Question

The main motive of this research is to explore and understand different pre-processing methods for text categorization and ultimately state best pre-processing method and classifier duo that facilitates efficient text categorization.

Report Structure

The subsequent parts of this paper is as follows:

[1]https://rapidminer.com/text-mining-customer-insights/
[2]https://trends.google.co.uk/trends/

- Related work section reports about the various research that have been carried out in the field of text analytics using machine learning techniques. It also gives information about the involvement of Ant colony optimization Algorithm across various classification problems.
- Methodology section contains step-by-step explanation of how this research was carried out.
- Implementation section gives detailed information of the procedures that were undertaken to achieve the results.
- Evaluation section explains the findings of the research with help of different use case studies.
- Finally conclusion section gives information about the final decision of the best preprocessing-classifier duo.

2 Background:

Term Document Matrix and Document Term Matrix:

Term Document Matrix helps to represent text documents in multi-dimensional space as vectors which is referred to as Vector space model. Here, each document is considered to be a vector in the model against the terms(words) that are present in that document. The terms present in each document are denoted as term frequency and represented as,

$$DTF = (tf1, tf2, tf3, ..tfn)$$

where tf1...tfn are frequency of terms in the document. Generally, a collection of d documents with t terms is represented in the form of matrix as t X d matrix which is the required Term document Matrix(TDM) (Bai and Manimegalai; 2010).

Document term Matrix(DTM) is generally a sparse matrix where in each row signifies the document information each column specifies for the term found in the document. (Silge and Robinson; 2017)

According to (Silge and Robinson; 2017)corpora is defined as object that contains raw strings which are tagged with additional meta-data and extra information.In simple words TDM and DTM are the transpose of each other.

Term Frequency-Inverse Document Frequency(TF-IDF):

TF-IDF works on the basis of assigning weights to the terms that are important to the documents(repetitive occurrence) and eliminates the terms which are less in number across the documents(Yaram; 2016).As in consideration with the weight of term or term frequency(tf) in the document can be given with more accurate value rather than the binary values (0 or 1). Usually common words possessing weak differentiating capability are assigned with higher weights than that of the words who hold higher discriminating power. To overcome this issue, a concept by name Inverse Document Frequency(IDF) is used which helps by means of considering the terms that have lower term frequency index in the corpus possess discriminating power. This analogy is called as zipfs law which states as Rank of a term in TF-IDF is inversely proportional to the frequency of

its appearance (Silge and Robinson; 2017) Generally, TF-IDF is represented as for term t its tf-idf weight w(t) is given as

$$w(t) = tf * log(N/df)$$

where,
tf=term frequency of document
df=document frequency
N=total number of documents (Chen et al.; 2016)

Since sentimental analysis(also known as opinion mining) has gained lot of popularity due its ability to extract subjective information more accurately than the traditional analysis methods[3]. Even to perform sentimental analysis the data can be converted either DTM, TDM or TFIDF due to its simple representation and good level of accuracy it delivers, but (Le and Mikolov; 2014) came up with solution more powerful than these methods called Doc2Vec approach. Since the ordering of words is lost in DTM, TDM or TFIDF methods and Doc2Vec approach solves this problem. As we know, sentimental analysis involves analyzing of collective words ranging from a single line of text to an entire document.

3 Related Work

Text classification using Machine Learning:

Machine learning techniques are being used extensively to achieve impacting results for Text analytics.

(Trstenjak et al.; 2014) developed a framework for text classification using KNN algorithm with use of TF-IDF pre-processing technique. This framework performed in a quite effective manner for classifying text documents pertaining to various categories. The findings from this research showed that the sensitivity of algorithm changes with change in type of documents it is given with. This indicates that even type of documents can impart its affect on the classifying capability of the classifier up to some extent. Similarly in the other research by (Bafna et al.; 2016) performed document clustering for E-mail classification with the help of k-means and Hierarchical clustering algorithms with TF-IDF as pre-processing method. This research was carried out in two phases were First case involved selection of better algorithm by virtue of the results which both of the clustering algorithms render when ran against a small chunk of original dataset. The algorithm with larger accuracy number was used to run against the remaining a part of the e-mail dataset and hence the results were calculated. Both of the clustering methods were found to work well with e-mail sorting or e-mail classification.

(Wang and Qian; 2008) implemented an algorithm which used Support vector machine(SVM) with Linear Discriminant Analysis(LDA) for categorizing text documents that automatically picks most repetitive terms from Bag of words(Corpus). Here, a high dimension data set was projected into a low dimension data set and then the algorithm was fed with low dimension dataset. Here SVM acted as classifying algorithm and LDA

[3] https://en.wikipedia.org/wiki/Sentiment_analysis

was used as dimension reduction tool. Findings from this research showed a better classifying performance than its competitors. This research was further extended by (Liu et al.; 2009) by use of additional performance metric called Feature enhanced smoothing method, which stated that particular terms that are not present in training corpus can help achieve good classification performance. The main concept behind this research was to consider the relativity between the terms particularly which are not present in train data but are present only in test data. By virtue of this method, the feature selection process was smoothed hence the name Feature enhanced smoothing method.

Under many situations, training classifier with data(documents) which are systematically labelled and organized turn out to be expensive, since it requires efforts to curate data in to such format. In order to tackle this scenario, (Shafiabady et al.; 2016) carried out a research to use unsupervised clustering method to train Support vector machine for classifying text documents. In this research, they used two approaches namely Self-organizing maps(SOM) and Correlation co-efficient to group text data which are unlabeled and then this grouped data was fed into SVM to train then test for categorizing the text documents. The findings for binary classification gave good results were as when the number of categories increased then there was drop in the accuracy level. But on the whole it illustrated that unsupervised clustering prior to SVM classification can turn out to be feasible whenever domain expert knowledge for labeling data is not available.

In a research by (AbuZeina and Al-Anzi; 2017) carried out classification of Arabic text using Linear discriminant analysis(LDA) also known as Fischer's LDA. The research included classification of corpus that contained around 2000 documents in Arabic language. The results from this research rendered that performance of semantic loss LDA was almost same as that of semantic rich singular value decomposition(SVD). Hence they concluded Linear discriminant analysis method was more suitable efficient in classifying Arabic text documents or in general LDA turns out to be effective in text mining.

(Onan et al.; 2016) carried out a research in a way of automatic keyword extraction which can be helpful in the ways of automatic procedures for text summarization, text classification, clustering as well as filtering. This research mainly concentrates on studying the statistical keyword extraction methods and running them against Machine learning and ensemble methods for classifying text data. And then finally comparing their performance. This research proved that Ensemble techniques along with utilizing keyword based representation of text documents will increase the scalability of classification schemes enhances predictive performance which is very important in the field of Text Analytics. In a research carried out by (Dadgar et al.; 2016) using TF-IDF on SVM for classifying Text documents pertaining to various datasets gave accuracy of 97.84percent of accuracy which was better than other classifiers.

Finally, (Fernández-Delgado et al.; 2014) carried out research using 179 classifiers across 17 families on 121 data sets to find out the best performing classifiers and the dependent attributes which act as driving force for that performance and found out that Random forest was the best performing classifier with 92.3percent accuracy.

By seeing from all of the above research, it strikes with a clear point that different machine learning approaches have performed well against different variants of data. Hence considering these results and collaborating them against different preprocessing procedures. Instead of considering one approach nearly all of them are considered and some are newly added because of the complexity involved in Text analysis to extract features and use them as guide to build an efficient Text categorization model. Since different pre-processing approaches are tested to their full potential against different classifiers

which makes this research on of its kind. As the different researches carried on in the past have been concentrating on increasing the classifier accuracy and there hasn't been made to concentrate equally on the preprocessing aspect as well. In that case the categorization capability of the above defined approaches can be enhanced if a sweet spot is found in terms of finding a perfect pre-processing method for a particular classifier. This research has been carried out in pursuit of filling that gap and defining with a near perfect classifier-preprocessing duo.

3.1 Ant Colony Optimization for classification:

Ant Colony optimization algorithm (Dorigo and Stützle; 2010) works in a way to assign each cases to a class out of different classes on the basis of some attributes called Predictor attributes, it can be illustrated as follows

IF $< conditions > THEN < class >$

The IF part has certain conditions and when a particular case/cases matches satisfies with the condition, then class associated with that satisfied condition is given out as result. This procedure of satisfying each condition with the rule is called as Ant Miner Rule (Parpinelli et al.; 2002)

According to (Wu et al.; 2012) Ant Colony Optimization(ACO) is a meta-heuristic algorithm that works in a similar fashion as the behavior of ants in the ant colony. ACO has been proved to be an effective optimization algorithm in place of combinatorial optimization algorithms like 0-1 knapsack, Travelling salesman problem etc. But however, applying the ACO algorithm for text classification comes down to the point of how to perform feature selection, hence (Nemati et al.; 2009) successfully carried out feature selection using ACO and Genetic algorithm(GA) in the context of protein function prediction. This research mainly concentrated to achieve a classifier which better in terms of search capability. Feature selection is a procedure which involves selection of subset of features from a large feature set in virtue of reducing feature space dimensionality to achieve better classification.(Tabakhi et al.; 2014) and this research concluded that ACO works very well to achieve feature selection in terms of tackling time-complexity problem.

In order to incorporate text classification using ACO (Meena et al.; 2012) carried out research to perform text categorization. This research mainly concentrates on the preprocessing part of text documents by means of Feature selection and came up with a very interesting conclusion. As the research says accuracy of the feature selection classifier increased with increase in the number of iterations the ant performed on that data. Feature selection approach has been widely used in variety of applications and one of the interesting application of feature selection using ant colony was done by (Kanan and Faez; 2008) for developing a facial recognition system. This research was carried out on ORL database to determine the facial recognition without priori information and the ACO performed well than other algorithms.

The main reason for inclusion of ACO in this research is because of its excellent performance in solving combinatorial problems which involved procedures to find shortest distance between source and destination. Basically, this process comes down to the process of classifying different available paths into shortest and longest paths. Using this property of ACO can yield good results in classifying text as well. Solving combinatorial problems and text classification can be related in a manner like, as the ACO classifies

the available paths as shortest and longest in the same way ACO can be used to classify text and assign it to category which is more relevant.

Another powerful aspect of ACO is that it provides with the provision of selecting the number of ants to be included in the classification procedure and number of iterations to be carried out to deliver the results.

4 Methodology

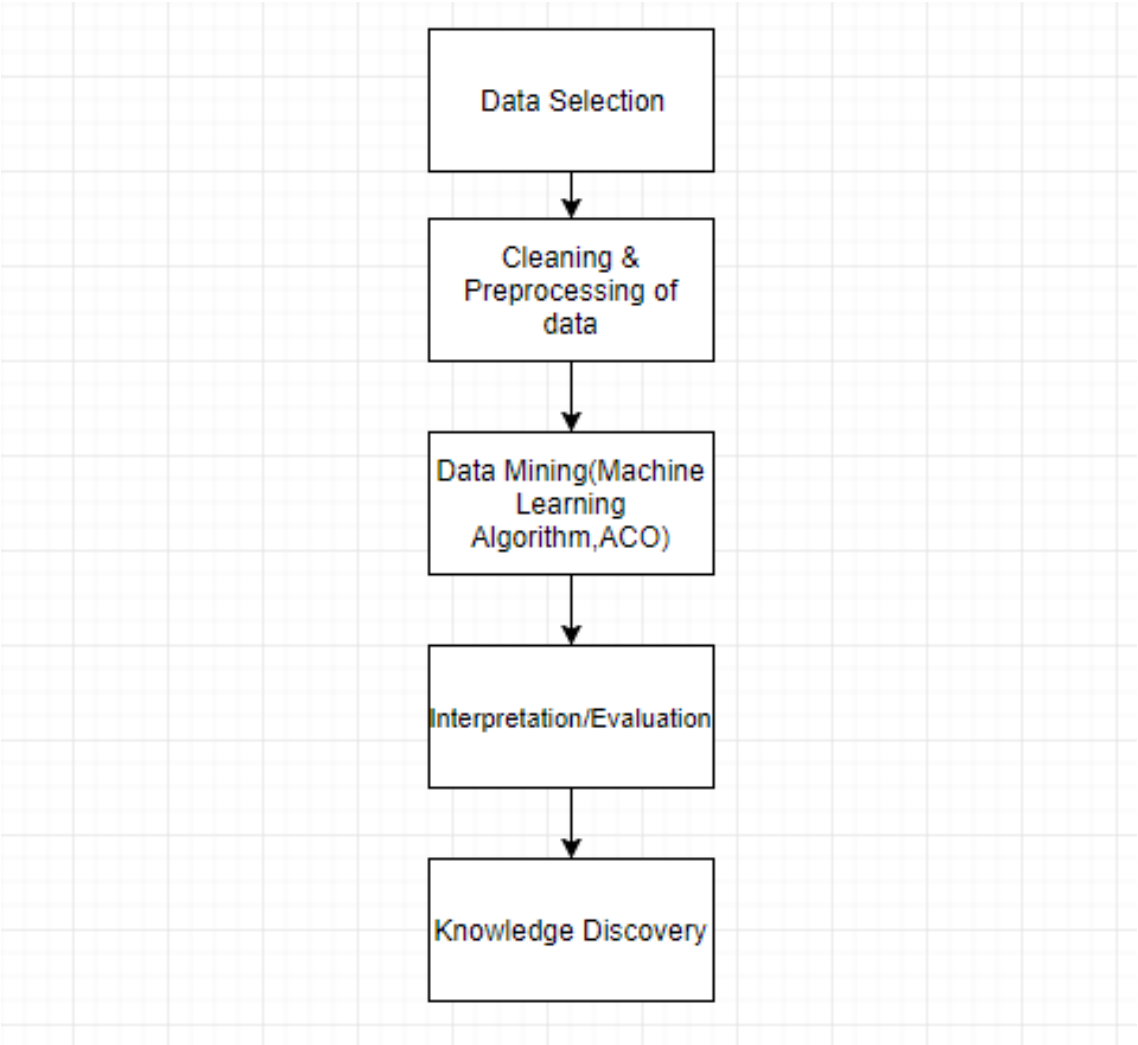

Figure 2: Instrumentation of KDD

4.1 Data Collection

Data collection is the initial and most important part, since this project involves classification of text data as its primary goal. BBC News data [4] is taken as the input data. Basically, this data consists of news headlines pertaining to different categories/Domains which include Sports, Politics, Entertainment, Technology and Business. In total there are 2225 documents.

[4] http://mlg.ucd.ie/datasets/bbc.html

The main reason to select this data is due to the variety of content it covers across all domains and sometimes it becomes a difficult task to determine the type of news. For instance, a news headline pertaining to sports category might not have the word sports in it and this becomes a problem for individuals if their knowledge in regard to sports is weak as they won't be recognize that headline as a sports headline. Figure 2 shows instrumentation of KDD.

4.2 Data Pre-processing

In this step, Data is cleaned and pre-processed in order to pass it through the classifiers. Cleaning of data in the terms of removing special characters, white spaces, punctuation, stop words etc. Since presence of these elements decreases the quality of corpus built. Then after the cleaning procedures the data is converted to corpus(bag of words). This corpus is then converted to vectors in the form of TDM,DTM and TF-IDF.

Antminer[5] is the ACO framework that is used in this research. This framework is written in java and takes input files in .arff format specifically. This is achieved using a package in R called Foreign[6]. The .arff file pertaining to all the categories of pre-processed data is fed into Antminer.

As discussed earlier, ACO creates rules from training data with conditions associates class(here news categories)with those rules and then checks the rules created against the test data. As the size of data increases, so does the number of rules and in addition to that since the whole procedure is done with 10-cross fold validation with large number of repetitions. In order to achieve all these tasks, it requires more computational resources.Hence antminer platform was set on cloud in Openstack platform.

4.3 Running Data against Machine Learning Algorithms and Ant Colony Optimization Algorithm

Once the data is pre-processed into different formats, they are made to run against the classifiers to obtain the different aspects of output, which will be classified text data in to different categories.

As discussed earlier regarding the research in regard to various classifiers processing capability of text data to achieve efficient categorization. In addition to that the classifier performance will also vary to the type of preprocessed data it is fed with.

4.4 Evaluation and Ranking of Results

Precision, Recall and F-score are the measures which are taken into consideration for evaluating the performance of Machine Learning classifiers. For ACO algorithm the performance is measured by the terms of accuracy. On the basis of all the results rendered, the classifiers are ranked on the basis of their performance(Best to Worst).These metrics give out the performance by the means of binary information in terms of True positives(TP), False Positive(FP), True Negatives(TN), False Negatives(FN) (Hassan et al.; 2016)

Accuracy is the most widely used measure to determine the performance of classifier as it the ratio of correctly predicted observations to that of total number of observations

[5] `http://www.aco-metaheuristic.org/aco-code/public-software.html`

[6] `https://cran.r-project.org/web/packages/foreign/foreign.pdf`

under consideration and this performance measure holds good data which is well balanced with equal number of false positives and false negatives.But for data with unequal number of data and with large number of classifiers to classify against is challenge. We are more interested in determining the true positives and looking for getting more true positives or correctly classified values which are indeed correct. Precision can be defined as ratio of correctly predicted positive values to that of total positive predicted values

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

Another performance measure is Recall which gives information about the data which were classified correctly over all the data that were actually correct in the data.Recall is the ratio of correctly predicted values to that all the data

$$Recall = \frac{TP}{TP + FN}$$

F-score is the weighted score of Precision and Recall, it can sometimes be mistaken as accuracy but it is way more of use than accuracy as it considers both false positives and false negatives and gives more accurate figure when used to evaluate data that has imbalanced data.

$$F.score = \frac{2 * (precision.recall)}{precision + recall}$$

4.5 Interpretation

The final results obtained from the evaluation phase is visualized using charts and graphs to get visual insights of the results.

5 Implementation

BBC News Dataset is taken which has different categories of News headlines. Figure 3 shows the architecture of the project implemented.

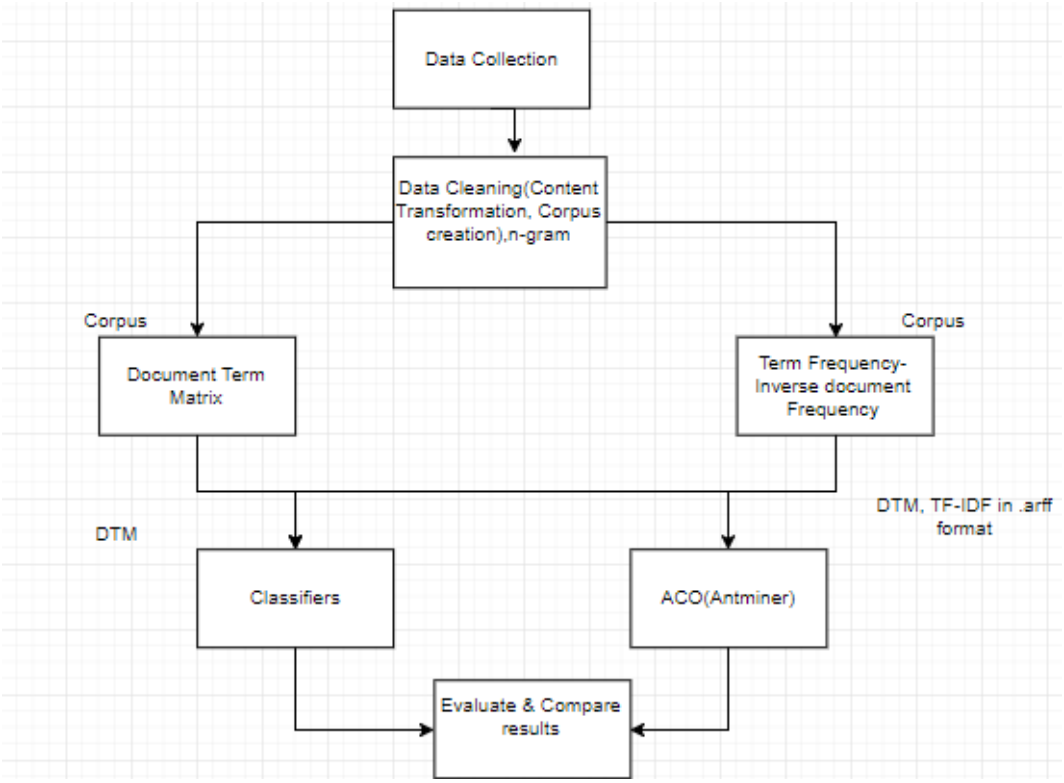

Figure 3: System Architecture

Once the data is collected, it is converted to a corpus and then subjected against content transformations like Removing white spaces, punctuation, end words, special characters etc. The cleaned and the simplest form of corpus is then subjected to different pre-processing strategies like TDM, DTM and TF-IDF. Different pre-processed data is then passed to different Machine learning classifiers as well as with ACO. When all the classifiers complete their execution then the results obtained are interpreted and then classifiers are ranked based on their performance(Best to Worst).

The next step in the procedure involves, the splitting of training data and labelling the categories manually. I can say this step in the whole project was the tedious part since the original data set was not in this format. Then the classifiers were implemented using the library RText Tools in R by (Collingwood et al.; 2013) to process the data. Figure 4 shows the data transformation strategy.

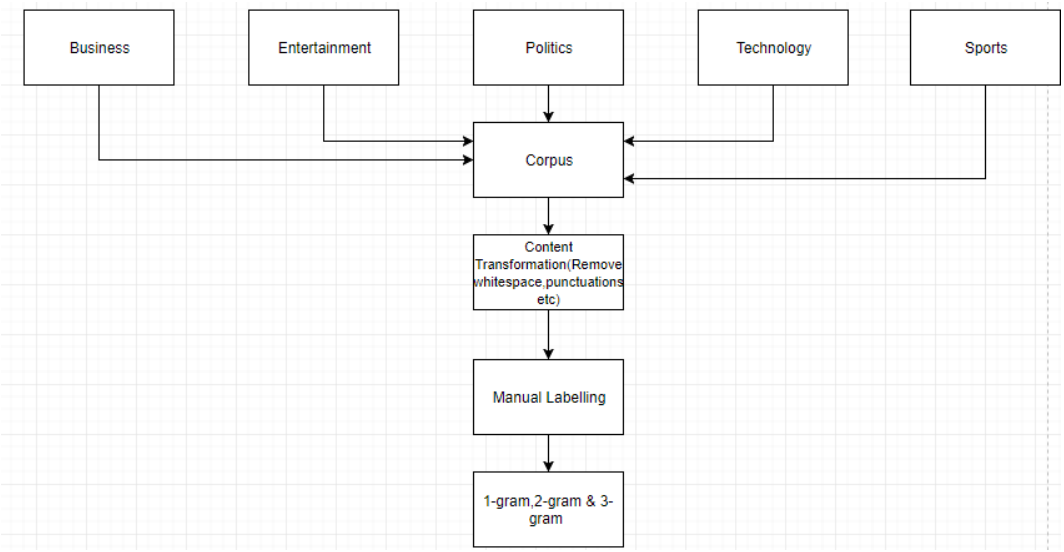

Figure 4: Data transformation strategy

Since there was no Feature subset available for the dataset, Feature selection was done by creating feature subset in different ways like Document term matrix, Term Document matrix and Term Frequency-inverse document frequency with Uni-grams, Bi-grams and Tri-grams. These pre-processed and cleaned data were ran against various classifiers like Support Vector Machine (Dimitriadou et al.; 2005) especially focuses on the features that are difficult to differentiate and works in way to find the best different between those features and come up with a best possible feature. SLDA since classification is a discriminative phenomenon and SLDA uses much powerful discriminative criteria to formulate the features.Bagging and Boosting are ensemble methods which implement n-learning algorithms from a single learning algorithm and gives with the average prediction value across all the learning algorithms, In addition to these both Bagging and Boosting perform well to manage Bias-Variance trade-off(Wang and Pineau; 2016).

Decision tree splits the dataset and generates classification model in an incremental manner. Since, Feature selection is an important aspect in classification we see the top nodes of the decision tree built on training data set are the most important feature on that dataset(Liu et al.; 2018).Neural Net interestingly works on DTM, TDM and TFIDF in a way to learn not only on the most repetitive but also learns on the words having assigned less weight or non-repetitive words (Prusa and Khoshgoftaar; 2016). Random Forest works in an efficient manner for large scale data and is quite effective in handling missing data or imbalanced data without compromising on the quality of classification. It also helps to understand the interactions among the variables which ultimately decide the best possible Feature set (Chaudhary et al.; 2016) (Liaw et al.; 2002)

GLMNET is comparatively easy to implement and manages well with data having large number of features and finally builds the Feature set (Korzeń et al.; 2013)(Friedman et al.; 2010), MAXENT is used because of its special property as it does not assumes the features present in the dataset are independent of each other and works in way to find the dependency among the features, since the features in text classification are words which are dependent on each other and ultimately combination of particular words leads to them getting classified to a particular category (Jurka; 2012).

In order to extract more relevant and effective feature set Unigram, Bigram and trigrams were used. However, there was no significant difference between the results

obtained between Bi-grams and Tri-grams. Figure 2 illustrates the resulting n-gram visualization of Business News category. We can see the frequency of most repetitive terms

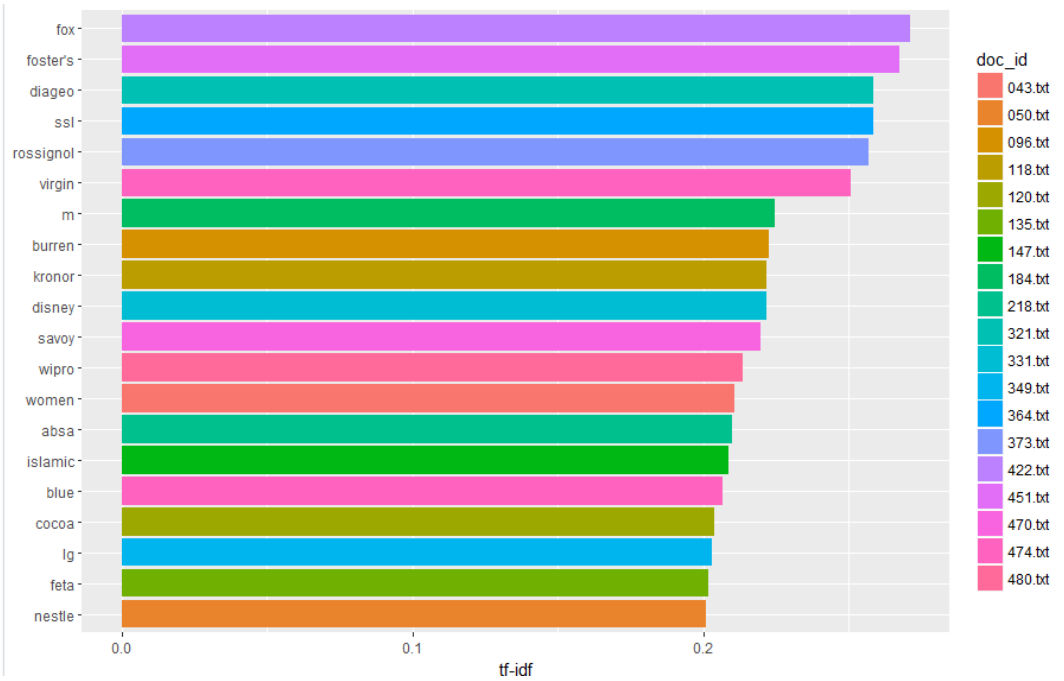

Figure 5: n-gram Analysis of Business category News

Finally, the results from classifiers, Antminer were all collected, ranked and then visualized to find out which particular combination of pre-processing procedure, classifier and categories yield best classified result.

Evaluation Strategy

Following are the evaluation strategy that have been used to evaluate the results: The subsequent parts of this paper is as follows:

- The data was split in to (70 : 30) split
- All the classifiers(Both Machine Learning and ACO) were ran with 10-cross fold validation to ensure better delivery of results.
- For ACO, there were in total 5 ants used with 5 iterations carried out to ensure proper training and testing.
- On a whole, the number of permutations carried out to arrive at the final results can be given as, For Machine Learning Algorithms,

  $9(Algorithms) * 3(preprocessingmethods) * 5(categories) * 10(cross-foldvalidation) = 1350$
- For ACO, the number permutations carried out can be given as,

  $1(Algorithm) * 3(preprocessingmethods) * 5(categories) * 10(cross-foldvalidation) * 5(iterations) = 750$

In total, Performance of Machine learning algorithms were measured in 1350 ways and performance of ACO were measured in 750 ways. In total BBC data set was evaluated in 2100 ways and the results were interpreted.

The subsequent section gives information about the Evaluation of the results found.

6 Evaluation

Table 1 show the performance of various classifiers against TDM in terms of Precision, Recall and F-score.

Feature selection performance of different classifiers against all of pre-processed data types with 10 cross-fold validation is determined by looking at the precision, Recall and F-score values delivered by the classifiers. From the table 1 we can see SVM, SLDA and RF were the best performers with f-value of 0.64 where as NNET, TREE and ACO gave f-value of 0.04,0.17 and 0.28 respectively when they were made to evaluate against TDM pre-processed data.

The figures implicate that when TDM is used as pre-processing method against the classifiers with 10 cross-fold validation, SLDA would be the better choice as classifier.

Algorithm	Precision	Recall	F-Score
SVM	0.65	0.63	0.64
SLDA	0.65	0.64	0.64
BOOSTING	0.59	0.57	0.58
BAGGING	0.36	0.37	0.36
RF	0.67	0.62	0.64
GLMNET	0.65	0.62	0.63
TREE	0.16	0.19	0.17
NNET	0.06	0.12	0.04
MAXENT	0.59	0.58	0.58
ACO	0.27	0.30	0.28

Table 1: Performance of classifiers against TDM

Table 2 shows the performance of various classifiers against DTM in terms of Precision, Recall and F-score.

From the table 2 we can see SVM, SLDA, RF and MAXENT were the best performers with f-value of 0.64 where as NNET, TREE and ACO gave f-value of 0.13,0.12 and 0.21 respectively when they were made to evaluate against DTM preprocessed data. Surprisingly, we saw MAXENT joined the group of best performing classifiers when it was made to perform with DTM which implies DTM favours MAXENT to perform well when compared to TDM.

The figures implicate that when DTM is used as pre-processing method against the classifiers with 10 cross-fold validation, RF would be the better choice as classifier.

Algorithm	Precision	Recall	F-Score
SVM	0.69	0.68	0.68
SLDA	0.69	0.66	0.67
BOOSTING	0.52	0.56	0.53
BAGGING	0.54	0.39	0.45
RF	0.71	0.73	0.71
GLMNET	0.66	0.64	0.65
TREE	0.10	0.17	0.12
NNET	0.15	0.12	0.13
MAXENT	0.69	0.59	0.63
ACO	0.20	0.23	0.21

Table 2: Performance of classifiers against DTM

Table3 shows Performance of classifiers against TF-IDF in terms of precision, Recall and F-score

From the table 3 we can see SVM,SLDA,RF and MAXENT were the best performers with f-value of 0.64 where as NNET, TREE and ACO gave f-value of 0.21,0.22 and 0.31 respectively when they were made to evaluate against DTM preprocessed data.

The values implicate that when TF-IDF is used as pre-processing method against the classifiers with 10 cross-fold validation, SLDA would be the better choice as classifier.

From the findings above, SLDA was found to be the best performer when TF-IDF was used as pre-processing method than other classifiers and pre-processing methods. In addition, the results revealed performance of all the classifiers were comparatively better with TF-IDF than the other two.

Algorithm	Precision	Recall	F-Score
SVM	0.65	0.66	0.65
SLDA	0.70	0.72	0.71
BOOSTING	0.48	0.50	0.48
BAGGING	0.51	0.37	0.42
RF	0.70	0.72	0.71
GLMNET	0.63	0.64	0.63
TREE	0.20	0.24	0.22
NNET	0.20	0.23	0.21
MAXENT	0.69	0.59	0.63
ACO	0.30	0.32	0.31

Table 3: Performance of classifiers against TF-IDF

since TF-IDF gave the best performance, Ensemble agreement was ran against the TF-IDF classified results.Basically, Ensemble agreement is carried out to determine the number of classifiers that predict the same result.

Since the main intent of this research is to find out the best pre-processing method that facilitates classifiers to perform a better classification task. As we noticed in the evaluation section above, TF-IDF was the aiding classifiers to perform better than the other two. By performing Ensemble agreement we focus on making a good procedure(TF-IDF) to a best procedure as to establish better classifier performance. It also acts as a

supporting factor to conclude TF-IDF as a better procedure.This gives the main reason to carry out Ensemble agreement on TF-IDF procedure. Table 4 gives information of the Ensemble agreement on TF-IDF

n	n-ENSEMBLE COVERAGE	n-ENSEMBLE RECALL
n≥ 1	1.00	0.77
n≥ 2	1.00	0.77
n≥ 3	0.98	0.78
n≥ 4	0.91	0.82
n≥ 5	0.78	0.85
n≥ 6	0.65	0.88
n≥ 7	0.46	0.90
n≥ 8	0.27	0.94
n≥ 9	0.13	0.98

Table 4: Ensemble summary for TF-IDF

For a 4-ensemble agreement, around 91percent of data was classified with 82percent accuracy.

All the above tests where run to classify the five categories of news dataset.

Performance of classifiers with varying categories

The findings were plotted on the graph by considering the top 2 performing classifiers and bottom 2 performing classifiers along with the results of ACO.

Classifier Performance with TDM

In this section, classifier performance is evaluated against different categories of data with TDM pre-processing method i,e classifier is subjected 2-category data through all 5-categories to see the variance in the classifier performance.

Figure 6 shows the classifier performance against different categorical data.

As we see from the figure we see, ACO was the best performing with 2-category data and on the other side was the worst performer too but with 5 categories to classify. The findings reflected major drop in the classification accuracy of ACO when the number of categories were increased. Whereas, RF was best performing overall with a slow and gradual decrease in the accuracy.

Other than ACO, RF performed well with steady drop in the accuracy level with increase in categories.

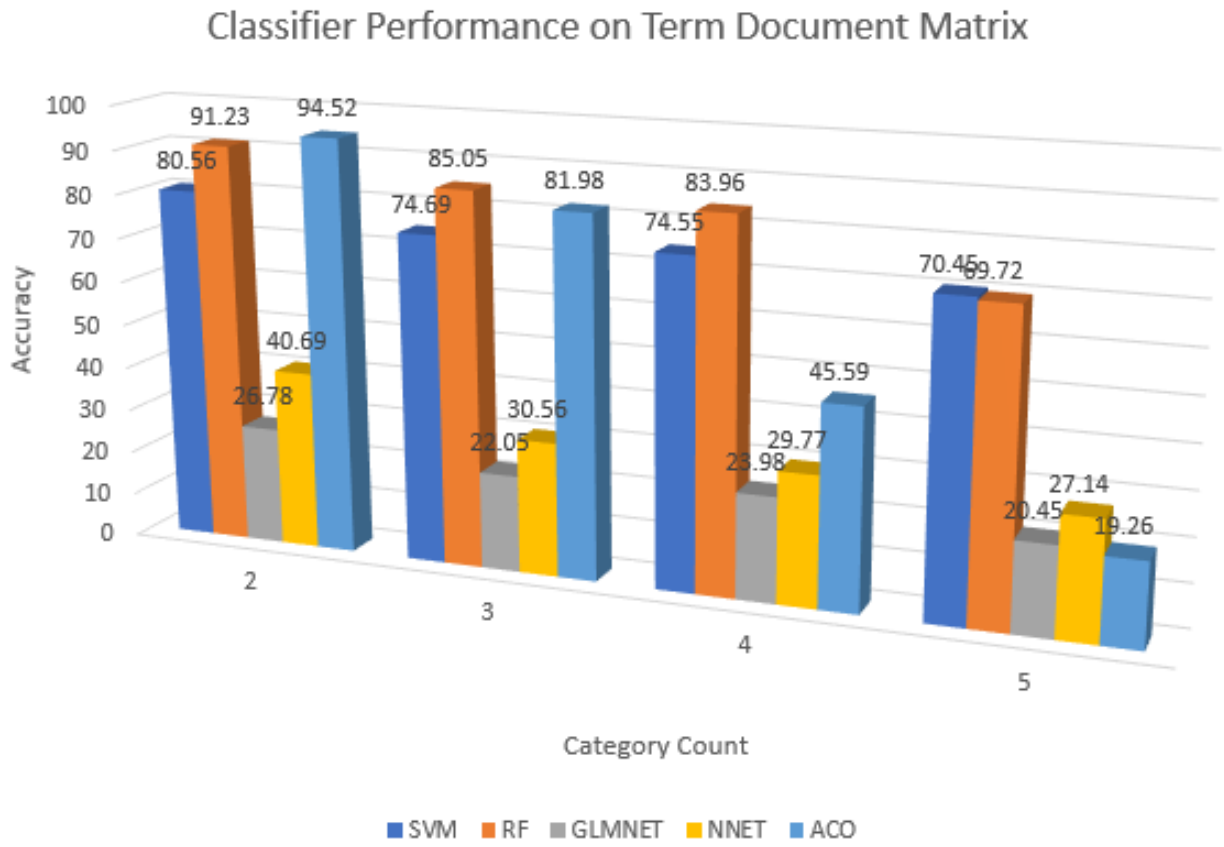

Figure 6: Classifier Performance against TDM with varying categories

6.1 Classifier Performance with DTM

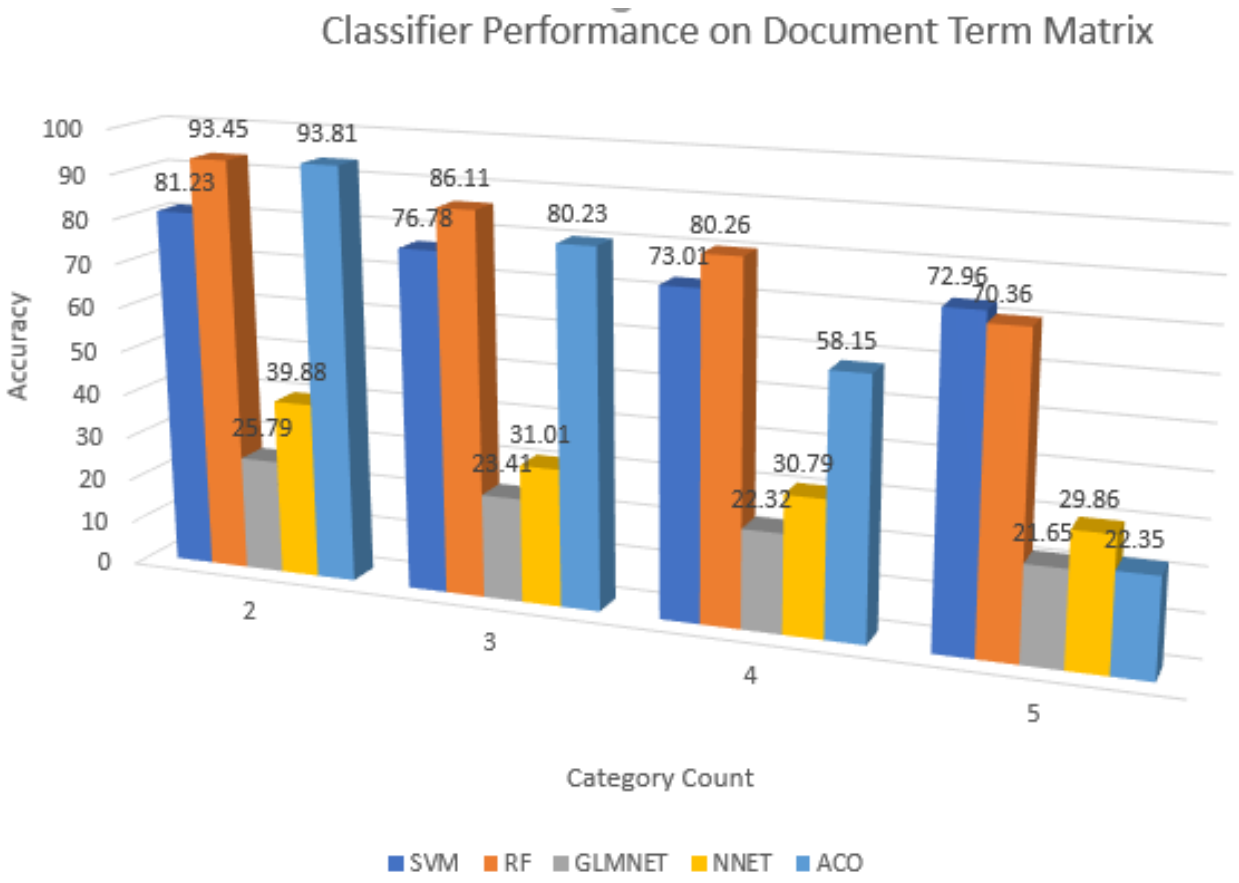

Figure 7: Classifier Performance against DTM with varying categories

In this section, DTM pre-processing method was selected with varying categories and then subjected to classifiers.

Results across all the categories reflected the same pattern of accuracy as TDM. ACO was the best performing for Bi-categorical data and then followed by RF. However, the worst performing classifiers of TDM performed better with DTM with higher accuracy rate.

6.2 Classifier Performance against TF-IDF with varying categories

In this final section of evaluation, TF-IDF procedure was selected against the classifiers for evaluation with varying number of categories. Even though the variance across the categories was same as the previous two methods but there were two interesting findings

- ACO performed better across all the categories as when compared to the performance of other two procedures.
- Even the performance of Machine learning algorithms were much better.

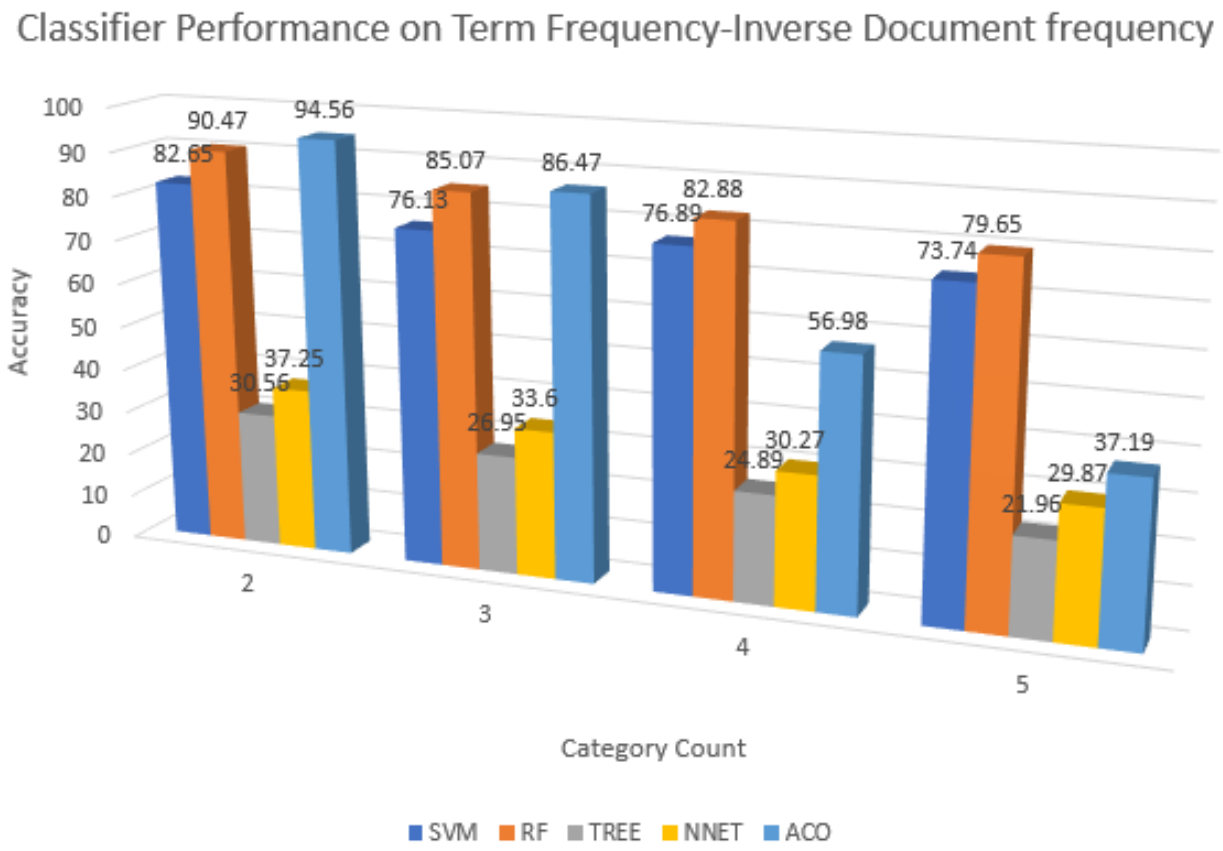

Figure 8: Classifier Performance against TF-IDF with varying categories

The findings which we saw in the above sections were much more informative and provided much better support towards stating TF-IDF as a better pre-processing method than TDM DTM. Even it gave an insight of how and when ACO performs best than other classifiers and when to avoid use of ACO. Talking about the overall classifier performance Random Forest stands on top of the list and SVM makes it second to that list.

7 Conclusion and Future Work

This research presented findings of an experimental study of various text pre-processing techniques and their impact on the classifier performance to achieve an efficient Text categorization model. In particular, three text pre-processing methods namely Document Term Matrix(DTM), Term Document matrix(TDM) and Term Frequency-Inverse Document Frequency(TF-IDF) were subjected against various classifiers like SVM, SLDA, RF, TREE, NNET, GLMNET, MAXENT, BOOSTING, BAGGING and ACO on BBC News dataset. The evaluation measures used were Accuracy, Precision, Recall and F-score. The results indicate that on a whole TF-IDF tends to produce better results than DTM and TDM with RF classifier. Also, ACO performed better against Bi-categorical data with TF-IDF as pre-processing strategy. When looking at both of the results it makes a much clear point about TF-IDF pre-processing procedure comparatively aids better results across all the classifiers.

Future work aims to compare performance of these pre-processing procedures against Unsupervised Machine learning algorithms and find out measures that enhance the classifier performance.

Acknowledgements

I would like to thank my Supervisor (Dr. Simon Caton) who has been a constant support and helped me in all possible ways to get this research done. I would also like to thank my parents for the support and motivation. My last acknowledgement is to all my friends who have been there with me throughout the whole time and supporting me through the hard times.

References

AbuZeina, D. and Al-Anzi, F. S. (2017). Employing fisher discriminant analysis for arabic text classification, *Computers & Electrical Engineering* .

Bafna, P., Pramod, D. and Vaidya, A. (2016). Document clustering: Tf-idf approach, *Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on*, IEEE, pp. 61–66.

Bai, V. M. A. and Manimegalai, D. (2010). An analysis of document clustering algorithms, *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on*, IEEE, pp. 402–406.

Chaudhary, A., Kolhe, S. and Kamal, R. (2016). An improved random forest classifier for multi-class classification, *Information Processing in Agriculture* **3**(4): 215–222.

Chen, K., Zhang, Z., Long, J. and Zhang, H. (2016). Turning from tf-idf to tf-igm for term weighting in text classification, *Expert Systems with Applications* **66**: 245–260.

Collingwood, L., Jurka, T., Boydstun, A. E., Grossman, E., van Atteveldt, W. et al. (2013). Rtexttools: A supervised learning package for text classification.

Dadgar, S. M. H., Araghi, M. S. and Farahani, M. M. (2016). A novel text mining approach based on tf-idf and support vector machine for news classification, *Engineering and Technology (ICETECH), 2016 IEEE International Conference on*, IEEE, pp. 112–116.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D. and Weingessel, A. (2005). Misc functions of the department of statistics (e1071), tu wien, *R package version* pp. 1–5.

Dorigo, M. and Stützle, T. (2010). Ant colony optimization: overview and recent advances, *Handbook of metaheuristics*, Springer, pp. 227–263.

Fernández-Delgado, M., Cernadas, E., Barro, S. and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems, *J. Mach. Learn. Res* **15**(1): 3133–3181.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent, *Journal of statistical software* **33**(1): 1.

Hassan, M. A., Malik, A. S., Saad, N. and Fofi, D. (2016). Evaluation metric for rate of background detection, *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*, IEEE, pp. 1–5.

Ittoo, A., Nguyen, L. M. and van den Bosch, A. (2016). Text analytics in industry: Challenges, desiderata and trends, *Computers in Industry* **78**: 96–107.

Jurka, T. P. (2012). Maxent: an r package for low-memory multinomial logistic regression with support for semi-automated text classification, *The R Journal* **4**(1): 56–59.

Kanan, H. R. and Faez, K. (2008). An improved feature selection method based on ant colony optimization (aco) evaluated on face recognition system, *Applied Mathematics and Computation* **205**(2): 716–725.

Korzeń, M., Jaroszewicz, S. and Klesk, P. (2013). Logistic regression with weight grouping priors, *Computational Statistics & Data Analysis* **64**: 281–298.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188–1196.

Liaw, A., Wiener, M. et al. (2002). Classification and regression by randomforest, *R news* **2**(3): 18–22.

Liu, D., Xu, W. and Hu, J. (2009). A feature-enhanced smoothing method for lda model applied to text classification, *Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference on*, IEEE, pp. 1–7.

Liu, Z.-T., Wu, M., Cao, W.-H., Mao, J.-W., Xu, J.-P. and Tan, G.-Z. (2018). Speech emotion recognition based on feature selection and extreme learning machine decision tree, *Neurocomputing* **273**: 271–280.

Martens, D., De Backer, M., Haesen, R., Vanthienen, J., Snoeck, M. and Baesens, B. (2007). Classification with ant colony optimization, *IEEE Transactions on Evolutionary Computation* **11**(5): 651–665.

Meena, M. J., Chandran, K., Karthik, A. and Samuel, A. V. (2012). An enhanced aco algorithm to select features for text categorization and its parallelization, *Expert Systems with Applications* **39**(5): 5861–5871.

Nemati, S., Basiri, M. E., Ghasem-Aghaee, N. and Aghdam, M. H. (2009). A novel aco–ga hybrid algorithm for feature selection in protein function prediction, *Expert systems with applications* **36**(10): 12086–12094.

Onan, A., Korukoğlu, S. and Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification, *Expert Systems with Applications* **57**: 232–247.

Parpinelli, R. S., Lopes, H. S. and Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm, *IEEE transactions on evolutionary computation* **6**(4): 321–332.

Prusa, J. D. and Khoshgoftaar, T. M. (2016). Designing a better data representation for deep neural networks and text classification, *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*, IEEE, pp. 411–416.

Shafiabady, N., Lee, L. H., Rajkumar, R., Kallimani, V., Akram, N. A. and Isa, D. (2016). Using unsupervised clustering approach to train the support vector machine for text classification, *Neurocomputing* **211**: 4–10.

Silge, J. and Robinson, D. (2017). *Text mining with R: A tidy approach*, " O'Reilly Media, Inc.".

Tabakhi, S., Moradi, P. and Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization, *Engineering Applications of Artificial Intelligence* **32**: 112–123.

Trstenjak, B., Mikac, S. and Donko, D. (2014). Knn with tf-idf based framework for text categorization, *Procedia Engineering* **69**: 1356–1364.

Vandierendonck, H., Murphy, K., Arif, M. and Nikolopoulos, D. S. (2016). Hpta: High-performance text analytics, *Big Data (Big Data), 2016 IEEE International Conference on*, IEEE, pp. 416–423.

Wang, B. and Pineau, J. (2016). Online bagging and boosting for imbalanced data streams, *IEEE Transactions on Knowledge and Data Engineering* **28**(12): 3353–3366.

Wang, Z. and Qian, X. (2008). Text categorization based on lda and svm, *Computer Science and Software Engineering, 2008 International Conference on*, Vol. 1, IEEE, pp. 674–677.

Wu, B., Wu, G. and Yang, M. (2012). A mapreduce based ant colony optimization approach to combinatorial optimization problems, *Natural Computation (ICNC), 2012 Eighth International Conference on*, IEEE, pp. 728–732.

Yaram, S. (2016). Machine learning algorithms for document clustering and fraud detection, *Data Science and Engineering (ICDSE), 2016 International Conference on*, IEEE, pp. 1–6.