# Comparison of text classification techniques

Srikanth Amudala, Kamal Maanicshah
Email: srikanth.amudala@mail.mcgill.ca , kamal.mathinhenry@mail.mcgill.ca

***Abstract*—One of the applications of machine learning is in the classification where model construction is utilized for text classification. The more the model learns from previous data, the more accurate it would perform. The ability of a classification model to achieve high accuracy in text mining is of significant importance. The complexity of text data unfavourable for most models, which is a major challenge, the training and testing data in the classification of data must be selected in a way that the model enjoys the most efficient learning from previous data and the highest accuracy in text classification. In this study, the Reddit dataset of comments, the models for predicting the categories are developed based on logistic regression, Support Vector Machine and Naive Bayesian methods and the accuracy of each model is evaluated. The best performance and accuracy, while it depends on the nature and complexity of the dataset.**

## I. Introduction

Text classification problems have been widely studied and addressed in many real-world applications over the last few years. Especially with Natural Language Processing (NLP) and text mining, many researchers are now interested in developing applications that leverage text classification methods. Most text classification and document categorization systems can be deconstructed into the following four phases: Feature extraction, dimension reductions, classifier selection, and evaluations. In this paper, we discuss the structure and technical implementations of text classification systems by the accuracy of the results using Naive Bayes, Logistic Regression, Support Vector Machines, Gradient Boosting classifier, k-nearest neighbours and Random Forest.

Any text classification algorithms have four different levels of scope. 1. Document-level: The relevant categories of a full document are obtained in Document level. 2. Paragraph level: In the paragraph level, the algorithm obtains the relevant categories of a single paragraph (a portion of a document). 3. Sentence level: In the sentence level, obtains the relevant categories of a single sentence (a portion of a paragraph). 4. Sub-sentence level: In the sub-sentence level, the algorithm obtains the relevant categories of sub-expressions within a sentence (a portion of a sentence). Our work in this paper focuses on the sentence level. The pipeline of classification is divided into the following :

### A. Feature Extraction

In general, messages and archives are unstructured informational collections. These unstructured content arrangements must be changed over into an organized component space as a major aspect of a classifier. The data should be cleaned to discard the non-useful characters and words, the data preprocessing is explained in detail in section 3. After the data has been cleaned, formal element extraction strategies are applied. The normal methods of highlight extractions used in this paper are Term Frequency-Inverse Document Recurrence (TF-IDF), Term Frequency (TF) and Word2Vec. We have used all the techniques and found TF-IDF and Term Frequency are more useful.

### B. Dimensionality Reduction

As content or report informational collections frequently contain numerous interesting words, information pre-preparing steps can be slacked by high time and memory multifaceted nature. A typical arrangement to this issue is just utilizing cheap calculations. Notwithstanding, in certain informational collections, these sorts of modest calculations don't execute just as anticipated. To maintain a strategic distance from the reduction in execution, numerous specialists like to utilize dimensionality decrease to lessen the time and memory intricacy for their applications. Utilizing dimensionality decrease for pre-preparing could be more productive than creating economical classifiers. In Section 3, we discuss more the dimensionality reduction that we have applied to our data in order to improve the efficiency and accuracy.

### C. Classification Techniques

Choosing the best classifier is the next step in the text classification pipeline where we need to fit the data into a model that gives us accurate results.

### D. Evaluation

The final part of the text classification pipeline is evaluation and understanding how a model performs is essential to the use and development of text classification methods. There are many methods available for evaluating supervised techniques. Accuracy calculation is the simplest method of evaluation but does not work for unbalanced data sets

## II. Related work

Numerous research publications have been published on text classification over the past decade. A detailed explanation of different classification techniques was discussed in [1]. A simple Bayesian logistic regression approach that uses a Laplace before avoid over-fitting to produce sparse predictive models for text data has been proposed in [2]. A Comparison of event models for Naive Bayes text classification was discussed in [3].

## III. Datasets and data preprocessing

The dataset we experiment with for this project is the Reddit comments data provided in the Kaggle competition. This happened to be a challenging dataset than expected as the correlation between the words and the respective classes were minimal. Also, there were some classes which had similar keywords like the ones related to gaming: 'league of legends', 'overwatch', 'GlobalOffensive', etc. Similar problems are experienced in sports-related classes like 'NBA', 'hockey', 'soccer', 'baseball', 'nfl', etc. To obtain the most relevant keywords from the comments, first, we have to remove the unnecessary punctuation and another formatting in the comments as they hold minimal information for classification. For example, some of the comments had links to websites which are of no use as they may vary for each comment. Also, some people use special characters in their writing which has is of no use as well. In addition to this, we also have to take care of stop words in the data. Stop words are those which might be present in abundance in text data and might not help in classification. Good examples for these types of words are I, you, and, is, are, etc. Another important problem in text pre-processing is that a certain word may appear in more than one form. This is similar to the fact that play, plays, playing, played, etc are all different forms of the wordplay. The process of removing these variations is known as stemming and lemmatization. Basically, our pre-processing steps can be listed as 1) Removing hyperlinks, 2) Remove Special characters, 3) Remove stop words, 4) stemming and lemmatization of data. The next step is to find the words which are important for the proper classification of data. We do this by using TFIDF scores. We form a bag of words model based on these results which are used as input to the models. For Binomial Naive Bayes classifier the inputs had to be binary, hence if the word is present in the comment it is noted as 1 and 0 if not. For the other models, we used count data. For the XLNET model, we did not use any pre-processing.

## IV. Models Used

### A. Support Vector Machine

Support vector machines (SVM) are a group of eager learning methods utilized in classification and regression problems. The SVM uses non-linear mapping to minimize empirically classification error and maximize the geometric margin [5]. The objective of this algorithm is to discriminate positive data from the set of negative data with the maximum border line in the area of features. With the assumption that, the categories are separable in a linear manner, hyper-planes with the maximum margin is developed to separate the categories. In the case where data are not separable in a linear manner, data is mapped in a larger space to separate them in linear manner [6] In SVM a given data is observed as a P-dimensional vector (or a list of P). In this method attempt is made to separate the points with a P-1 dimensional hyperplane. This process is named linear discrimination. There exist various hyperplanes that separate the data. Based on this technique SVM are also referred to as Maximum Margin Classifiers. The algorithm searches for the maximum distance between the two closest points of each class. Initially the SVM determines a hyperplane that has the largest fraction of points of one class on the same plane. A hyperplane is also determined for the next class forming two parallel hyperplanes making it the best case for our data set which has 20 different classes.

### B. Logistic Regression

Logistic regression is practical in many areas such as text classification since it is a linear model with several parameters equal to the dimensionality of the dataset. suppose a text classifier, $y = f(x)$, from a set of training examples $D = (x_1, y_1), ..., (x_n, y_n)$ that we want to learn. For text categorization, the vectors $x_i = [x_{i1}, ..., x_{id}]^T$ comprise transformed word frequencies from documents It models the conditional probability as:

$$w_{k+1} = w_k + \alpha_k \sum_{i=1}^{n} x_i \left(y_i - \sigma\left(w_k^T x_i\right)\right) \tag{1}$$

$$\text{with} \quad \sigma(x) = \frac{1}{1 + \exp\left(-x\right)}$$

For a book classification issue, $p(y)$ relates to the likelihood that the $i_{th}$ report has a place with the class. The choice of whether to relegate the classification can be founded on looking at the likelihood with a limit or, all the more for the most part, given augmenting the normal adequacy [8]

### C. Bernoulli Naive Bayes

The Naive Bayes Algorithm is a form Bayesian classifiers which are statistically structured. Based on the Bayesian theorem, the Na¨ıve Bayes assumes that the presence of a particular attribute of a class is unrelated to the presence of another attribute [7] It assumes all attributes to be independent giving it the name Na¨ıve. The classifier also assumes that no hidden or latent attributes influence the prediction process. The the algorithm used the probabilities of each attribute belonging to each class to make a prediction [7]

Let assume $C$ is a random class in a data set of $m$ classes. Let $X$ be a given tuple of random variables denoting observed attribute values. Given $X$ the classifier will predict that $X$ belongs to the class having the highest posterior probability conditioned on $X$. Posterior probability refers to the probability of a specific tuple based on a hypothesis or condition. The classifier predicts that $X$ belongs to class $Ci$ if and only if $P(C_i|X) > p(C_j|X)$ for $1 < j < m$ and $x, j \neq 0$ Thus the class $C_i$ for which $P(C_i - X)$ is maximized known as the maximum posterior hypothesis [7].

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{2}$$

where $P(C_i|X)$ is the posterior probability of class given predictor, $P(C)$ is prior probability of class, $P(X_{c_i})$ likelihood the probability of predictor given class and $P(X)$ prior probability of predictor.

### D. K-Nearest Neighbour(KNN)

In this method, it is assumed that all the samples are points in real n-dimensional space and the neighbour s are determined based on standard Euclidean distance. K is the number of neighbours. The objective in this algorithm is to find the best class for a given data, in a way that the distance with its class members is the shortest. One of the methods in KNN algorithm is data validation through the application of k-fold cross-validation.

### E. Random Forest

Random forests technique is an ensemble learning method for text classification the main idea of random forest is generating random decision trees. In contrast with other techniques such as deep learning, random forests are very easy to train for text data sets but quite slow to make predictions. The number of trees in the random forest must be decreased to achieve a faster structure, as more trees in the forest increase the amount of time in the prediction stage.

### F. XLNET

XLNET is a deep learning framework which is good to model the contexts between the words in a sentence. For example, New York should be considered as two words that occur together instead of being separated. This is done by the BERT algorithm as well. However, the BERT algorithm induces masks in the input data which is undesirable. Besides, the dependencies between the masked words are ignored as well. These shortcomings are overcome by XLNET by integrating a transformer architecture which finds the relation between the words in a sentence by looking at the complete sentence in its entirety. Due to this architecture, we do not use any preprocessing for the data.

## V. Results

The data is preprocessed using the preprocessing steps that were discussed in Section 3. After the Term Frequency and the Term Frequency-Inverse Document Recurrence (TF-IDF), we have around 68 thousand features which would consume a lot of memory and computational power. To tackle this problem, we have implemented different dimensionality reduction techniques and found PCA to be more useful. We were successfully able to reduce the dimensions from 68000 to 2000. Binomial Naive bayes classifier was not efficient with the features we have selected for the data set and resulted in poor accuracy although it was able to give better accuracy in a sample data set we have used from Kaggle. The accuracy of the other models with the output from PCA was not very good and staggered at around 40 to 50%. So we tried a different approach by building a machine learning pipeline and passing the TF and TF-IDF as a parameter to the model and feed the preprocessed data into the model for training. We have implemented different models to see which one works better with the data set we have and found Multinomial Naive Bayes classifier to be the better among the rest. More details about the accuracy comparison are given in TABLE I. But the Multinomial Naive Bayes Classifier resulted in an accuracy of 56% which is not very ideal when compared with the TA baseline that was provided. So we have tried implementing the model with XLNET [4] which resulted in a much better accuracy when compared with all the other models.

TABLE I: Model Comparison

| **Head** | ***Model name*** | ***Accuracy*** |
|---|---|---|
| 1 | XLNET | 57% |
| 2 | Multinomial Navie Bayes Classifer | 56% |
| 3 | Logistic Regression | 54% |
| 4 | SVM | 54% |
| 5 | Random Forest | 47% |
| 6 | KNN | 42% |
| 7 | Binomial Naive Bayes Classifier | 8% |

## VI. Conclusion

We demonstrated in our experiments with 6 different text classification models and found that XLNET yields consistently higher effectiveness than the other algorithms. Multinomial Naive Bayes was almost similar to that of XLNET results but we have observed that the training time for XLNET was more when compared with the other models.

## References


[1] Text classification algorithms: A survey. Kowsari, Kamran and Jafari Meimandi, Kiana and Heidarysafa, Mojtaba and Mendu, Sanjana and Barnes, Laura and Brown, Donald

[2] Large-Scale Bayesian Logistic Regression for Text Categorization. Alexander Genkin, David D Lewis and David Madigan.

[3] A Comparison of Event Models for Naive Bayes Text Classification. Andrew McCallum, Kamal Nigam.

[4] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, Quoc V. Le: XLNet: Generalized Autoregressive Pretraining for Language Understanding. CoRR abs/1906.08237 (2019)

[5] Classification of diabetes disease using support vector machine, Kumari, V Anuja and Chitra, R

[6] Data mining: concepts and techniques. Han, Jiawei and Pei, Jian and Kamber, Micheline

[7] Micheline KAMBER a Jian PEI. Han, Jiawei

[8] Evaluating and optimizing autonomous text classification systems. Lewis, David D and others.