TEXT CLASSIFICATION ON NEWS ARTICLES

NG KANG WEI

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION
TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2020

TEXT CLASSIFICATION ON NEWS ARTICLES

NG KANG WEI

RESEARCH PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY UNIVERSITY OF MALAYA, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF DATA SCIENCE

2020

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: (I.C./Passport No.:   )

Registration/Matric No.:

Name of Degree:

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Field of Study:

 I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

 Candidate's Signature   Date:

Subscribed and solemnly declared before,

 Witness's Signature   Date:

Name:
Designation:

TEXT CLASSIFICATION ON NEWS ARTICLES

ABSTRACT

An abstract must not exceed 500 words, typed in a single paragraph with double- spacing, and written in Bahasa Malaysia and English language. A maximum of five (5) keywords should also be listed below the abstract.

Keywords: Keyword, keyword, keyword, keyword.

ABSTRAK

Ini merupakan abstrak dalam Bahasa Melayu (satu perenggan).

ACKNOWLEDGEMENTS

Thanks guys. I owe you many.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1   Introduction

As optical character recognition technology advances, large number of physical documents are made electronically available and many more articles are created and available online. The information contain within these documents would need to be extracted, analyzed, stored and made searchable so that others can make use of it. Natural language processing (NLP) methods are needed to analyze the content or the sentiment in the text.

Document classification is one of the NLP method that categorize the text into different topics or categories. This classification would be helpful to future researchers who want to find some topic from the text. The researchers could just focus on the category they are interested in rather than skimming through all the documents to obtain the intended information.

The technology breakthrough in recent years, machine learning algorithms, processing power of processors have been a boost in NLP field. With the breakthroughs, there has been an advancement of the methods used in NLP with artificial intelligence without the need of intervention of domain experts.

There are several approaches to the document classification problem, multilabel where each document can belong to several categories or classification, where each document can only belong to one category. Within machine learning methods, there is clustering which is an unsupervised learning method or the supervised learning approach. This shall focus in the the direction of classification rather than multilabel and clustering.

In document classification most of the algorithms used vector space model to represent the unstrucutured textual data. (Ababneh, Almanmomani, Hadi, El-Omari, & Alibrahim, 2014). This vector space model represent the sequence of the textual features and their

weight, it is easy to implement and provide uniform representation for documents. However, it has a drawback, it represent all the words in the documents, the dimension of the vector would be huge. This huge vector space model would impact the performance of the machine learning tasks. (Moldagulova & Sulaiman, 2018). Therefore, this study would focus on the dimension reduction on vector space model on document classification.

1.1.1 Problem Statement

Term vectors is one of the most commonly used document representation algorithms, but dimension of the feature space can too large and the vectors can be too sparse. (Moldagulova & Sulaiman, 2018)

1.1.2 Research Objectives

1. To identify a document representation algorithm that is optimized to extract the features from news articles

2. To investigate the machine learning (ML) algorithm used in document classification and apply the most suitable algorithm.

3. To evaluate the performance of the document represenation and document classification algorithm.

1.1.3 Research Questions

1. Which dimension reduction algorithm is optimized for news article?

2. How complexities of the features influence the accuracy?

3. Which is the best machine learning algorithms in document classification?

1.1.4 Research Motivation

In the age of big data, the amount of data generated and collected is growing at an explosive rate. Much of the data generated and collected is in the form of unstructured

text. The value contained within the text cannot be extracted and be useful to us without natural language processing (NLP). Document classification is one of the pillars in natural language processing.

In document classification, the unstructured text would be given a label or multiple label dependent on the method used. These labels would make the data more meaningful and searchable. Users can search for a topic just by selecting text with the particular label rather than performing a manual word search on all the text document.

Bag of words is the most commonly used document representation method in document classification. Bag of words would produce a vector space model of the textual data representation. The dimension of this vector space model would be big because of the amount of words in the documents. This huge dimension of vector space model would decrease the performance of the document classification algorithms.

With dimension reduction algorithms, the dimension of the vector space would be decreased and the performance of the machine learning algorithms would be increased. However, the effect of the different dimension reduction algorithms might have a different effect on the performance of the machine learning algorithms.

This study would investigate the effect of the dimension reduction algorithms on the performance of the machine learning algorithms.

1.1.5   Research Significance

This ressearch would classify news articles into different categories. In order to do that, first the features have to be extracted from the documents. The features might need to be compressed. Then the features are used to train machine learning models. After that, the trained machine learning models are validated and tested to evaluate its performance.

The document representation method is the most commonly used bag of words approach, Term Frequency - Inverse Document Frequency (TF-IDF). In this approach, the documents

are converted into vector space models. Due to the large amount of words in the documents, the vector space would be large and sparse, this is known as the curse of dimensionality problem. This large and sparse vector space would be an obstacle to document classification, machine learning models accuracy would be impacted due to the large vector space.

By applying dimension reduction algorithms to the vector space model, the vector space and sparsity of the vector could be reduced. With the reduced vector, the accuracy of the machine learning models would be increased. However, which of the dimension reduction algorithms would perform best for document classification on news articles? This study would try to answer this question by studying which of the dimension reduction algorithm is most suitabled and applying it on a dataset.

1.1.6 Expected Outcome

1. A prototype document classification application with 80

2. A dimension reduction algorithm is applied on the extracted features of the documents

3. A best suited machine learning algorithm is applied on the document classification application

4. Evaluate the accuracy of the document classification application

CHAPTER 2: LITERATURE REVIEW

2.1   Introduction

In text classification proposed in this study, there are 3 stages namely document representation, dimension reduction and classification. For document representation the chosen method is Term Frequency-Inverse Document Frequency (TF-IDF) which is based on the bag of words method. TF-IDF provide a measure of weight or importance to the words. The value of TF-IDF estimate the amount of information provided by each word in its document. (Arroyo-Fernández, Méndez-Cruz, Sierra, Torres-Moreno, & Sidorov, 2019) The value of TF-IDF increases proportionally to the number of times a word appears in a document but is offset by the frequency of the word in the corpus. This means that if a word appears for many times in a document but it also appears for many times in many other documents in the corpus, then it is not an important word. TF-IDFThe formula for TF-IDF is shown below:

$$TFIDF = tf \times \log_e \frac{N}{df} \qquad (2.1)$$

where:

tf = the number of times a word appears in a document

N = the total number of documents in the corpus

df = the number of documents that contain the word

The remaining parts are dimension reduction and classification, the methods in these 2 parts would be evaluated and chosen.

2.2 Dimension Reduction

2.2.1 Single Value Decomposition (SVD)

Single value decomposition is one of the most commonly used dimension reduction algorithms. It generalizes a complex matrix with many dimensions into a matrix of lower dimension via an extension of the polar decomposition. SVD detects the part of the data that contains the maximum variance in a set of orthogonal basis vectors. The data with the maximum variance would be the most prominent features of the data. (Sweeney et al., 2014)

Latent Semantic Analysis (LSA) a technique applied in natural language processing that apply SVD in its process. SVD is applied in LSA to transform the features by dropping the least significant values in the matrix thus reducing the dimensions of the matrix. (Karami, 2017)

Even though SVD has been applied in LSA and in other fields, it has been found that SVD is not as efficient as principal component analysis (PCA) and has a few drawbacks. SVD can extract the prominent features of the matrix but it does not help in reducing the sparsity of the matrix. In some of text clustering algorithms that used SVD for dimension reduction only find SVD useful when the features are redundant. (Narhari & Shedge, 2017)

2.2.2 Nonnegative Matrix Factorization (NMF)

Nonnegative matrix factorization (NMF) is a multiplicative updating solution to decompose a nonnegative temporal-frequency data matrix into the sum of individual components. The sum of individual components are calculated from a nonnegative basis matrix. (Chien, 2019)

NMF and its variant have found to be applied in many fields such as feature extractions, segmentation and clustering, dimension reduction and others. However, the nonnegativity constraints of NMF proved to be problematic when the data matrix is not strictly non-

negative. The semi-NMF relaxes the non-negativity constraint of NMF so that the resulting matrix has mixed signs. (Allab, Labiod, & Nadif, 2017)

Researchers who performed experiments to compare the performance of SVD, NMF and PCA have found that NMF do not perform as well as PCA. SVD and NMF achieved a similar accuracy in the experiment. This might due to both SVD and NMF are dependent on matrix decomposition technique while PCA is dependent on eigenvalue decomposition. (Mohamed, 2019)

NMF might be more suitabled for clustering as it took the least amount of time in clustering the data compared with SVD and PCA. (Mohamed, 2019). This advantage of NMF over SVD and PCA is negligible in this study since this study would focus on classification rather than clustering.

2.2.3   Principal Component Analysis (PCA)

Principal component analysis is one the state of the art dimension reduction algorithm. The main purpose of PCA is to project data samples from high dimensional space into low dimensional space by linear transformation while preserving the original data features as much as possible. (Ma & Yuan, 2019) In other words, PCA is used to emphasize the variations in the data, bring out the important features in the data.

PCA is no stranger to the text classification field. It has been applied to different languages of text classification other than English. Label Induction Grouping (LINGO) a technique used in categorizing Indian Marathi language text document. PCA is applied in LINGO performed better than SVD as PCA extract the features better and has less loss of information. (Narhari & Shedge, 2017)

In another research on text classification on Arabic text and English text, it is also found that PCA outperformed SVD and NMF. The researchers found that PCA yields better result in terms of accuracy and normalized mutual information. The advantage of PCA is

that after transforming the matrix, the important features vector are orthogonal to each other. PCA also has a whiting transform that reduce noises in the data which in turn boost the performance and the accuracy of the machine learning algorithm.(Mohamed, 2019)

A variant of PCA which is in the form of a tree structure has been applied on the dimension reduction on sentiment analysis. The technique is called tree-structured multi-linear principal component analysis (TMPCA). TMPCA can retain the sentence structure and word correlations. (Su, Huang, & Kuo, 2018). However, this is a novel technique and PCA has been proven to be effective enough to handle the amount of data in this study.

2.2.4 Summary

The 3 dimension reduction or matrix decomposition algorithms above are considered blind source separation (BSS) methods, unsupervised learning algorithms. Its performance might not be as good as a deep learning algorithm such as Word2vec but deep learning algorithm's performance is dependent on the the scale of the data. Deep learning algorithm can only perform well when there is a lot of data. In our study, the data might not be sufficient to use a deep learning algorithm, thus the above methods are reviewed. Out of the 3 dimension reduction algorithms reviewed above, PCA seems to be the most promising in the field of text classification. Therefore, PCA would be chosen as the dimension reduction algorithm in this study.

2.3 Document Classification

2.3.1 Support Vector Machine

Support vector machine is a machine learning algorithm that construct a hyper plane to separate the examples into different classes. It has been proven to be very effective in dealing with high dimensional data. (Shinde, Joeg, & Vanjale, 2017). It is also proven to produce dramatically best results for topic modelling in experiments with the Reuters dataset.

(Dumais, Platt, Heckerman, & Sahami, 1998). Various issues need to be considered when applying SVM in document classification, the processing of the data, which kernel to use, and the parameters of SVM. A variant of SVM, called one-class SVM which is trained only with positive information has been used in document classification. (L. M. Manevitz & Yousef, 2002). The authors experimented with different kernels of SVM (linear, sigmoid, polynomial, and radial basis) with different type of document representation method (binary representation, frequency representation, TF-IDF representation, and Hadamard representation). The best result (F1 score of 0.507) is achieved with binary representation, feature length 10 and with linear kernel function.

In another research, the researchers apply SVM in the classification on web document instead of news or ordinary text document. The document representation method used in this research is vector space model, just the nouns term in the web pages. The researchers experimented with different SVM kernels and varying the size of the training sets. Expectedly, the precision, recall and accuracy increased as the size of the training set increase. Linear kernel achieved the best result out of the various SVM kernels, a classification accuracy of 80% is achieved. (Shinde et al., 2017).

SVM is relatively new compare to others algorithm in the field of document representation. It is not very efficient with large number of observations and it can be tricky to find an appropriate kernel for the problem.

2.3.2   k-Nearest Neighbours (kNN)

kNN is a classification machine learning algorithm that classify objects based on the closest training examples in the feature space based on a similarity measure. It is a simple and effective classification, as it only need 3 prerequisites. The 3 prerequisites are training dataset, similarity measure and the value of k which is the number of closest neighbours to be considered.

kNN needs minimal training, it only needs to plot the training examples into a feature space. kNN has been applied in document classification before, it is found that kNN take significant longer time to classify a document into a topic. This is because kNN uses all the features of the data to compute the distance. Since the authors are using term vector space document representation method, the dimension of the feature space is high, thus the more time is needed for kNN to compute all the distance between the test object with the training objects. Other than the time taken to compute the distance, the k value is another obstacle in kNN algorithm. In a high dimensionality feature space and the points are not evenly distributed, the k value is hard to be determined.

To overcome the problems mentioned above, the authors applied term vector space reduction method, divide the document feature matrix into parts. Term vector space reduction reduces the sparsity of the document term matrix by removing the features less appeared in the corpus. By reducing the term vector space, a slight deterioration in the classification accuracy but the time cost is dramatically reduced. kNN still achieved an accuracy of 92.7% but the time taken reduced from 53 minutes to 11 minutes. (Moldagulova & Sulaiman, 2018)

2.3.3 Neural Network

Neural network has a resurgence in recent years as there is a breakthrough in the neural network as Geofrey Hinton (et al.) discovered a technique called Contrastive Divergence that could quickly model inputs in a Restricted Boltzmann Machine (RBM). RBM is a 2-layer neural network that model the input by bouncing it through the network. This process is less computationally complex than backpropagation. (Hinton, 2002).

Currently, neural network is applied in deep learning to solve various problems, document representation is one of them. Ranjan (et al.) applied Lion Fuzzy Neural Network on document classification. The researchers used WordNet ontology to retrieve

the semantic of the words, and then added the context information onto it, thus the features obtained are semantic-context features. The classification part is performed by Lion Fuzzy Neural Network, which is a variant of Back Propagation Lion (BPLion) Neural Network that includes fuzzy bounding and Lion Algorithm. The neural network model used is trained incrementally. It achieves a higher accuracy than Naïve Bayes and some variant of the Lion Neural Network. (Ranjan & Prasad, 2018)

Other than the modified neural network shown above, a simple feed-forward neural network is also efficient in document classification. By using the Hadamard product as document representation method, a simple neural network also can achieve a good classification accuracy in document classification compare to Naïve Bayes, kNN, and SVM. (L. Manevitz & Yousef, 2007)

2.4   Conclusion

From the review of dimension reduction algorithms, PCA performed best compared to SVD and NMF. In addition, PCA performed well in text classification field.

In machine learning algorithms for text classification, all 3 machine learning algorithms reviewed above would be applied. One of the objectives of this study is to investigate the performance of different machine learning algorithms in text classification. The same dataset would be used to train 3 models and the performance would be evaluated.

CHAPTER 3: RESEARCH METHODOLOGY

3.1   Introduction

This section would illustrate process flow to carry out the experiments to fulfill the aforementioned objectives of this research. The following flow charts would illustrates the steps taken in the several experiments.

3.2   Text preprocessing process flow

This flow chart above illustrates the preprocessing process. The news articles articles dataset would be in the form of raw text. There are some preprocessing to be done before features can be extracted from the text to build the text classification models. First, the raw text would be read into a dataframe or a data structure for ease of access and processing.

In the raw text files, there are some lines that are not relevant for text classification purpose, these lines would be removed to reduce the noise in the dataset. All the text in the dataset would be converted to lowercase for ease of processing. There are some words in the text that are useful in human speech but do not convey meanings in text analysis. These words are stop words, these words have to be removed as well. After stop words removal, the text should contain only the words that are vital to text analysis but there would be some punctuations and symbols in the text. These punctuations and symbols have to be removed as well to make the text cleaner and less noisy.

After that, there is an important step, lemmatization. Lemmatization would convert most of the words in the text to their root form which is known as a lemma. This would reduce the noise in the data by transforming similar words into a single word. The alternative to lemmatization would be stemming. Stemming would be faster than and need less processing power than lemmatization but stemming has a drawback against

```
┌─────────────────────────────────┐
│            Raw text             │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│     Read the file into dataframe │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│     Remove some redundant lines │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│     Change the text to lowercase │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│        Stop words removal       │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│       Punctuation removal       │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│          Lemmatization          │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│        Output to CSV file       │
└─────────────────────────────────┘
```
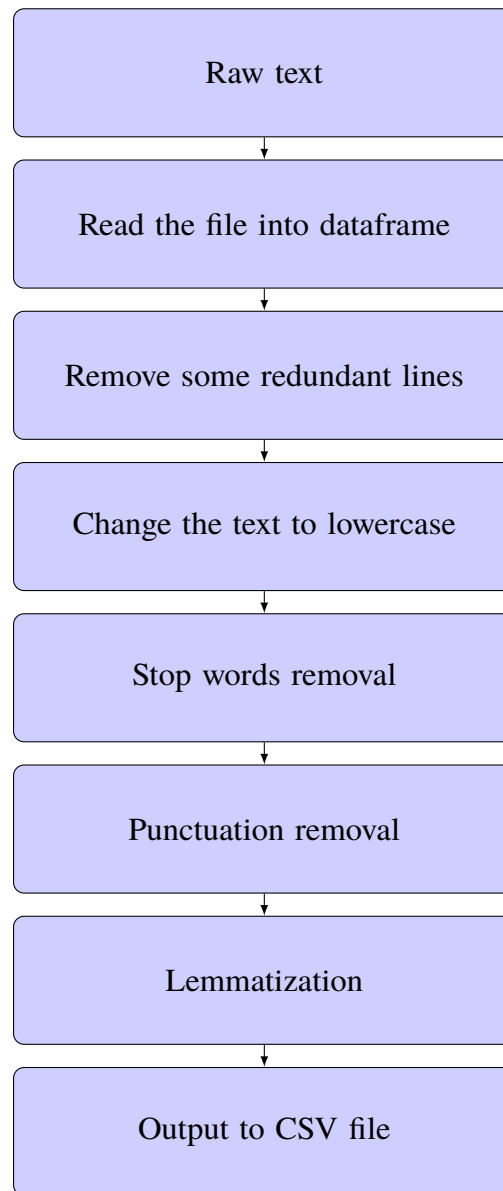
Figure 3.1: Overall process flow

lemmatization. The result of stemming may not be a real word because stemming just chop off the ends of the words thus the resulting words may not be a real words. On the other hand, lemmatization uses vocabulary and morphological analysis of words to remove the inflectional endings only and resulting the root form of words. (reference needed)

3.3   Overall Process Flow

The process flow chart above is the overall process for the few experiments. As mentioned above, the news articles dataset has to be preprocessed before it can undergo feature extraction and train machine learning models.
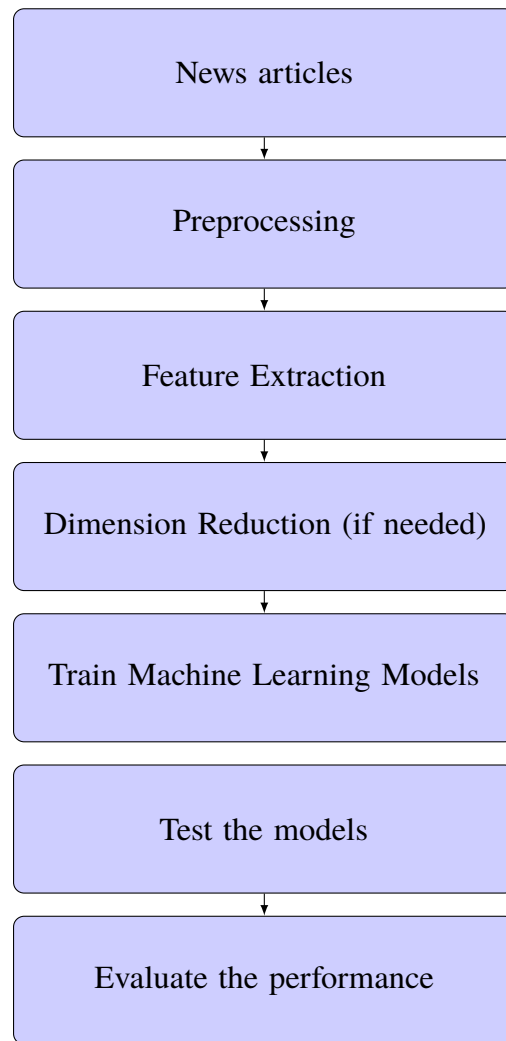
Figure 3.2: Process flow of preprocessing the text dataset

After preprocessing, the resulting data would be used in several experiments with a few subtle differences. The differences in the experiments would be in the feature extraction stage and the dimension reduction stage.

3.4 The experiments

The experiments that has been conducted in this research is as follows:

1. Term frequency

2. Term frequency with naive dimension reduction

3. Term frequency with SVD

4. TF-IDF

5. TF-IDF with naive dimension reduction

6. TF-IDF with SVD

Basically, there are 2 feature extraction method used in the experiments, namely term frequency and term frequency - inverse document frequency (TF-IDF). Each of the feature extraction method would be tested with and without dimension reduction algorithm.

There are 2 dimension algorithms involved, one is a naive method, which means that the features or columns lesser than a certain value would be removed. In other words, words or terms that do not appear much in the dataset would be removed. Another dimension reduction algorithm is single value decomposition (SVD). This method would retain the essence of the data, the part of the data with maximum variance.

After feature extraction and dimension reduction (if needed) are applied, the resulting features would be used to train machine learning models. There are 3 machine learning algorithms chosen in these experiments namely k-nearest neighbour (kNN), support vector machine (SVM), and neural network (NN). All 3 of the machine algorithms would be applied to all the different resulting features and the accuracy scores would be evaluated.

3.5   Conclusion

With the method mentioned above, the experiments would be carry out and the results would be recorded. The results would be compared and the differences would be analysed.

CHAPTER 4: RESULTS AND DISCUSSIONS

4.1   Introduction

The results of the experiments are shown in this section. The differences in the accuracy and performance of each of the methods would be discussed and analysed. Hopefully, the results and discussions would be able to answer the research questions posted in the beginning of this research.
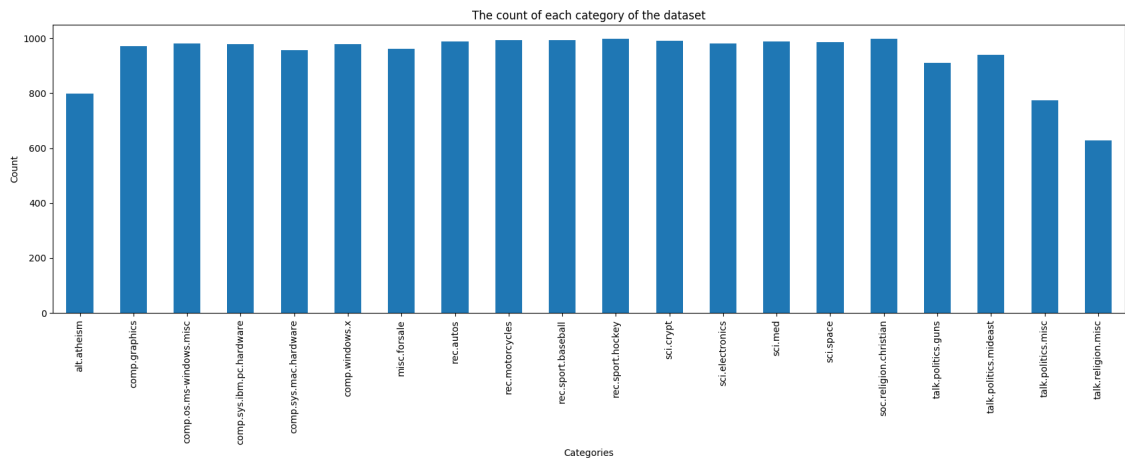
4.2   The dataset

Figure 4.1: The count of each category in the dataset

18790 rows, 15032 for training 3758 for testing details about the 20news dataset, the category distributions, a table, a chart

4.3   Term frequency

ML	no of features	accuracy	time taken (s)
kNN	8000	0.28	3.65
SVM	8000	0.81	5.27
NN	8000	0.84	91.82

Table 4.1: Term frequency

Term frequency is one of most common feature extraction method. It convert all the words in the dataset into a matrix where each column is a word and the rows are the number

of times each word appear in the text. Each row in the matrix is a document. This vector space model representation of the words would result in a sparse matrix since each of the documents only contains a subset of all the words in the whole dataset.

The number of features of the term frequency is limited to 8000, this is because of the memory constraint of the machine if it is unlimited the resulting matrix would be of bigger size and would have the probability of running out of memory while processing the matrix.

In the results shown above, NN achieved an accuracy of 0.84 which is the best accuracy out of the 3 classification algorithm but it is also the one that takes the longest to process. kNN took the least time to process but has the lowest accuracy score. Performance wise SVM is the best, it only took 5s to process, which is 80s less than NN and has comparable accuracy with NN.

reason of KNN accuracy low [reference] [https://arxiv.org/abs/1011.2807] kNN accuracy is low in this scenario is possibly due to the feature matrix is sparse. The data points are few and far in between. kNN classify a new data point based on the distance of the new data point with the nearby data points. Since the other data points are few and sparse, the classification of the new data points would be biased. kNN is designed for low dimensional data therefore it doesn't perform well in large and sparse data.

4.4   Term frequency with naive dimension reduction

The feature extraction method used in this experiment is same as the experiment above but dimension reduction method is applied to the resulting matrix before training. The dimension reduction algorithm used in this experiment is a naive one which means that the columns with the least occurrence of word are removed.

The parameter value is just an integer value passed to the dimension reduction function implemented. It has an inversely proportional relationship with the number of features. The larger the parameter value, the more columns would be removed and the number of

ML	parameter value	no of features	accuracy	time taken (s)
kNN	100	2530	0.27	4.16
kNN	500	478	0.32	4.88
kNN	1000	173	0.34	5.24
SVM	7	15854	0.83	6.5
SVM	10	12705	0.83	6.58
SVM	100	2530	0.76	6.16
NN	10	12705	0.87	116.78
NN	100	2530	0.80	48.08
NN	500	478	0.65	61.87

Table 4.2: Term frequency with naive dimension reduction

features would decrease.

In the result table above, each of the classification algorithm has different parameter value, this is determined by trial and error and the results shown are the best.

In this experiment, kNN still has the worst performance but its accuracy increases slightly as the number of features decreases and the time taken increases as well. SVM and NN have the same trend in this experiment, the accuracy and time taken decrease as the number of features decreases. This is because as the number of features decrease, there is less data to process thus the speedup and due to the decrease in training data, the model cannot correctly predict the test data.

In the scenarios where SVM and NN have more features than the experiment with just term frequency, both of the classification algorithms have better accuracy. These 2 classification algorithms are efficient and able to process features in a large and sparse vector space.

4.5   Term frequency with SVD

In this experiment, the feature extraction method used is still term frequency but the dimension reduction algorithm changed. Instead of reducing the dimension by the least term frequency, SVD is used. SVD would retrieve the features with maximum variance in

ML	no of features	accuracy	time taken (s)
kNN	2000	0.38	216.59
kNN	4000	0.31	482.66
SVM	2000	0.78	172.58
SVM	4000	0.80	370.43
NN	2000	0.79	91.13
NN	4000	0.8	220.76

Table 4.3: Term frequency with SVD

the data.

The number of features shown in the table above is the number of columns in the resulting matrix after applying SVD.

Similar with the trend in the previous experiment (term frequency with naive dimension reduction), the accuracy of kNN increases slightly when the features decreases but the resulting accuracy is still far from satisfactory. The accuracy of SVM and NN also have the same trend with the previous experiment, the accuracy decreases slightly when the number of features decreases.

As expected the time taken to process the features decreases as the number of features decreases. However, in comparison with the time taken in the previous experiments, the time taken in this experiment is astoundingly high. SVD would be the culprit, dimension reduction comes at a cost which is processing power. To reduce or compress the features into a small dimension, a lot of calculation would be involve, this would take both time and processing power.

The resulting accuracy is not as good as just with term frequency. This is expected because the number of features decreased. The reduction in features may save some memory space but in order to achieve that more processing power and time would be needed.

ML	no of features	accuracy	time taken (s)
kNN	8000	0.76	3.71
SVM	8000	0.87	2.39
NN	8000	0.88	55.82

Table 4.4: TF-IDF

4.6　　TF-IDF

A different approach in feature extraction is applied in this experiment. Instead of term frequency which only take the frequency of each word into account, the feature extraction method used here in TF-IDF. The main difference of this feature extraction method is that besides term frequency it also take the rarity of the words into account. If a term or word appear in high frequency but in many documents, this word may not be of importance or a meaningful feature. If a word appear rarely and only in a few documents, this word would have high importance and would be meaningful feature of the few documents.

With TF-IDF, kNN can achieved a satisfactory accuracy score of 0.76 even though the number of features in the resulting matrix of TF-IDF is the same with term frequency which is 8000. The vector space model of TF-IDF would not be as sparse as that of term frequency which is kNN is more suitable to be applied.

NN is still provide the highest accuracy score of 0.88 but the time taken also the longest at 55.82s.

SVM achieved an accuracy of 0.87 which is just 0.01 shy of what achieved by NN and the time taken is the lowest among the 3 which is 2.39s.

In comparison to the few experiments with term frequency, the accuracy and time taken of the 3 classification model improves. This prove that TF-IDF is a better feature extraction method for text.

ML	parameter value	no of features	accuracy	time taken (s)
kNN	50	4221	0.74	4.00
kNN	100	2530	0.71	4.11
kNN	500	478	0.49	5.15
SVM	50	4221	0.86	2.58
SVM	100	2503	0.83	2.63
SVM	500	478	0.66	2.94
NN	50	4221	0.85	38.80
NN	100	2530	0.83	34.28
NN	500	478	0.64	77.12

Table 4.5: TF-IDF with naive dimension reduction

4.7   TF-IDF with naive dimension reduction

Similar with the experiment with term frequency, naive dimension reduction is applied to the vector space model generated from TF-IDF. The trend over all the 3 machine learning models when the number of features decreases, the accuracy of the model decreases slightly and the time taken increases. The amount of information has become lesser, less information is available to train a comprehensive model, therefore the accuracy decreases.The dimension reduction even though is a naive one and less computing intensive still increases the time taken compared with just with TF-IDF. The more reduction is performed, the time taken would increase as well.

The accuracy achieve with TF-IDF with naive dimension reduction is still passable with kNN at 0.74 when the features reduced from 8000 to 4221. However, when the number of features reduced to 478, the accuracy is just at a measly 0.49 which is not satisfactory. SVM and NN also achieved comparable accuracy at 0.86 and 0.85 respectively in comparison with TF-IDF when the number of features are almost halved.

From the results of this experiment, it can be deduced that with naive dimension reduction, the number of features and memory needed to store the vector space model is reduced. With this reduced number of features, the classification models can still achieve comparable performance but would be at a slightly decreased capacity.

4.8 TF-IDF with SVD

ML	no of features	accuracy	time taken (s)
kNN	2000	0.58	208.64
kNN	4000	0.77	457.16
SVM	2000	0.86	69.63
SVM	4000	0.87	189.32
NN	2000	0.84	87.12
NN	4000	0.85	215.73

Table 4.6: TF-IDF with SVD

Finally, the last experiment is TF-IDF with SVD dimension reduction. Similar with experiments above with SVD, each of the machine learning models would be tested with 2 set of vector space model, each with different number of features, namely 2000 and 4000. To put it in perspective, the number of features without reduction is 8000.

When the number of features are at 4000 which is halved, kNN still can produced an accuracy of 0.77 which is similar with what is achieved with TF-IDF without dimension reduction. Same goes to SVM and NN, at 4000 features, the accuracy are quite similar with TF-IDF without dimension reduction. When the dimension of the vector space model is reduced to 2000, the accuracy across 3 of the classification models dropped. kNN being the most drastic, its accuracy dropped to 0.58 while SVM and NN dropped to 0.86 and 0.84 respectively, which is slightly worse than before but it is still satisfactory.

There is an odd trend on the time taken for all 3 classification models though. The time taken in this experiment is much higher than the experiment of TF-IDF without dimension reduction and TF-IDF with naive dimension reduction which is expected. SVD reduction would take more time as more calculation is performed. However, the time taken decreased when the further reduction is done, kNN take 457s to reduce the vector space model from 8000 to 4000 but just 208s to reduce the vector space model from 8000 to 2000. Keep in mind that the time recorded here includes the time taken for the classification model

to predict the test dataset as well. This trend appear in SVM and NN as well. SVM took 189s to reduce 8000 to 4000 but 69s to reduce 8000 to 2000 while NN took 215s to reduce 8000 to 4000 and 87s to reduce 8000 to 2000.

The decrement in time could be explained with the reduction of features, the time taken to process them also reduced therefore it provided a speedup. This speedup with the reduction of features came at a cost though, which is accuracy. For SVM and NN, the reduction in accuracy is meagre and would be logical to trade the slight accuracy with the speedup but in kNN the trade off would be lopsided.

CHAPTER 5: CONCLUSION

After conducting the experiments and analyse the results, the research find that...

answer the research questions and achieve the objectives of course

objectives:

1. to identify a document representation algorithm that is optimized

2. to investigate ML and find the most suitable

3. to evaluate the effect of dimension reduction has on document representation and document classification model performance

questions:

1. which dimension reduction algorithm is optimized

2. how complexities of the features influence the accuracy

3. the best ML

REFERENCES

Ababneh, J., Almanmomani, O., Hadi, W., El-Omari, N., Alibrahim, A. (2014, 02). Vector space models to classify arabic text. *International Journal of Computer Trends and Technology (IJCTT*, *7*, 219-223. doi: 10.14445/22312803/IJCTT-V7P109

Allab, K., Labiod, L., Nadif, M. (2017, Jan). A semi-nmf-pca unified framework for data clustering. *IEEE Transactions on Knowledge and Data Engineering*, *29*(1), 2-16. doi: 10.1109/TKDE.2016.2606098

Arroyo-Fernández, I., Méndez-Cruz, C.-F., Sierra, G., Torres-Moreno, J.-M., Sidorov, G. (2019). Unsupervised sentence representations as word information series: Revisiting tf–idf. *Computer Speech and Language*, *56*, 107-129. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0885230817302887` doi: https://doi.org/10.1016/j.csl.2019.01.005

Chien, J.-T. (2019). Chapter 5 - nonnegative matrix factorization. In J.-T. Chien (Ed.), *Source separation and machine learning* (p. 161 - 229). Academic Press. Retrieved from `http://www.sciencedirect.com/science/article/pii/B9780128045664000176` doi: https://doi.org/10.1016/B978-0-12-804566-4.00017-6

Dumais, S., Platt, J., Heckerman, D., Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on information and knowledge management* (pp. 148–155). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/288627.288651` doi: 10.1145/288627.288651

Hinton, G. E. (2002, August). Training products of experts by minimizing contrastive divergence. *Neural Comput.*, *14*(8), 1771–1800. Retrieved from `http://dx.doi.org/10.1162/089976602760128018` doi: 10.1162/089976602760128018

Karami, A. (2017, Nov). Taming wild high dimensional text data with a fuzzy lash. In *2017 ieee international conference on data mining workshops (icdmw)* (p. 518-522). doi: 10.1109/ICDMW.2017.73

Ma, J., Yuan, Y. (2019). Dimension reduction of image deep feature using pca. *Journal of Visual Communication and Image Representation*, *63*, 102578. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1047320319301932` doi: https://doi.org/10.1016/j.jvcir.2019.102578

Manevitz, L., Yousef, M. (2007). One-class document classification via neural networks. *Neurocomputing*, *70*(7), 1466 - 1481. Retrieved from `http://www.sciencedirect.com/science/article/pii/S092523120600261X` (Advances in Computational Intelligence and Learning) doi: https://doi.org/10.1016/j.neucom.2006.05.013

Manevitz, L. M., Yousef, M. (2002, March). One-class svms for document classification. *J. Mach. Learn. Res.*, *2*, 139–154. Retrieved from `http://dl.acm.org/citation.cfm?id=944790.944808`

Mohamed, A. (2019). An effective dimension reduction algorithm for clustering arabic text. *Egyptian Informatics Journal*. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1110866518301579` doi: https://doi.org/10.1016/j.eij.2019.05.002

Moldagulova, A., Sulaiman, R. B. (2018, Oct). Document classification based on knn algorithm by term vector space reduction. In *2018 18th international conference on control, automation and systems (iccas)* (p. 387-391).

Narhari, S. A., Shedge, R. (2017, Dec). Text categorization of marathi documents using modified lingo. In *2017 international conference on advances in computing, communication and control (icac3)* (p. 1-5). doi: 10.1109/ICAC3.2017.8318771

Ranjan, N. M., Prasad, R. S. (2018). Lfnn: Lion fuzzy neural network-based evolutionary model for text classification using context and sense based features. *Applied Soft Computing*, *71*, 994 - 1008. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1568494618304046` doi: https://doi.org/10.1016/j.asoc.2018.07.016

Shinde, S., Joeg, P., Vanjale, S. (2017, 09). Web document classification using support vector machine. In *2017 international conference on current trends in computer, electrical, electronics and communication (ctceec)* (p. 688-691). doi: 10.1109/CTCEEC.2017.8455102

Su, Y., Huang, Y., Kuo, C. . J. (2018, Aug). Efficient text classification using tree-structured multi-linear principal component analysis. In *2018 24th international conference on pattern recognition (icpr)* (p. 585-590). doi: 10.1109/ICPR.2018.8545832

Sweeney, E., Vogelstein, J., Cuzzocreo, J., A Calabresi, P., Reich, D., M Crainiceanu, C., T Shinohara, R. (2014, 04). A comparison of supervised machine learning algorithms and feature vectors for ms lesion segmentation using multimodal structural mri. *PloS one*, *9*, e95753. doi: 10.1371/journal.pone.0095753

APPENDIX A: MANUALS, TECHNICAL SPECIFICATIONS, DOCUMENTATIONS, EXAMPLE SCENARIOS

APPENDIX B: TRY

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.