Coursework CMT207 – Text Categorisation **DRAFT DATE: 06/ 04 / 22**

## Abstract

#

#

#

#

## Introduction

In recent years the quantity of digitised text has increased substantially, therefore allowing for more complex analysis and processes, including the application of machine learning methods.

Gaining deeper understanding of text has always been a key interest of researchers. One such area is text categorisation. Text categorisation is the automatic assignment of natural language (i.e. human language) to a predetermined categories based upon their content. (Lewis, 1994).

This project classifies a document of English language text into 1 of 20 distinct categories having been trained on the well-known “20 Newsgroups” dataset. This data set contains over 20,000 text documents spread across 20 distinct categories, for example, “motorcycles”, “hockey” and “religion”. These plain-text data take the form of email messages about the specified topic.

To perform this, the data was appropriately pre-processed (including vectorising and tokenising). We then trained the data on multiple models.

use simple models and more advanced neural network models to categorise documents into the relevant categories.

After running the model on the testing dataset found XX to be the best performing model havinf an accuracy of xx, and representing an xxxx performance suggesting the model is able to determine

## Literature Review

## Pre-processing

## Descriptive Analysis

The 20 Newsgroup data set is comprised of approximately 20,000 distributed amongst 20 categories, each with approximately 900 documents in each. The dataset provided had already been split into training (60%) and testing (40%) datasets.A summary of the dataset is shown in Table 1

For effective categorisation the training data should be distributed approximately equally between each group.

Of the training data set there were between 377 and 600 documents with a corpus ranging from 35,869 to 101,427 (with the exclusion of stop-words) for each category. The proportion of the cleaned corpus has a range from 2.6% (rec.autos) to 9.2% (alt.atheism). This is within an order of magnitude so we consider the data to be distributed equally. However, these differences may be a feature we would want to incorporate into our model.

Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Category** | **Training Number of Documents** | **Training Word Count** | **Average Words per Doc** | **Proportion of Corpus (%)** |  |
| **0** | rec.sport.hockey | 600 | 67385 | 112.3 | 4.9 |  |
| **1** | soc.religion.christian | 599 | 56353 | 94.1 | 4.1 |  |
| **2** | rec.motorcycles | 598 | 50948 | 85.2 | 3.7 |  |
| **3** | rec.sport.baseball | 597 | 48298 | 80.9 | 3.5 |  |
| **4** | sci.crypt | 595 | 42867 | 72.0 | 3.1 |  |
| **5** | sci.med | 594 | 79624 | 134.0 | 5.8 |  |
| **6** | rec.autos | 594 | 35869 | 60.4 | 2.6 |  |
| **7** | sci.space | 593 | 55166 | 93.0 | 4.0 |  |
| **8** | comp.windows.x | 593 | 49193 | 83.0 | 3.6 |  |
| **9** | comp.os.ms-windows.misc | 591 | 54828 | 92.8 | 4.0 |  |
| **10** | sci.electronics | 591 | 72390 | 122.5 | 5.3 |  |
| **11** | comp.sys.ibm.pc.hardware | 590 | 101427 | 171.9 | 7.4 |  |
| **12** | misc.forsale | 585 | 51022 | 87.2 | 3.7 |  |
| **13** | comp.graphics | 584 | 78629 | 134.6 | 5.7 |  |
| **14** | comp.sys.mac.hardware | 578 | 81608 | 141.2 | 5.9 |  |
| **15** | talk.politics.mideast | 564 | 92699 | 164.4 | 6.8 |  |
| **16** | talk.politics.guns | 546 | 85512 | 156.6 | 6.2 |  |
| **17** | alt.atheism | 480 | 125966 | 262.4 | 9.2 |  |
| **18** | talk.politics.misc | 465 | 88406 | 190.1 | 6.4 |  |
| **19** | talk.religion.misc | 377 | 54878 | 145.6 | 4.0 |  |
| **Total** |  | 11314 | 1373068 |  |  |  |
| **Average** |  | 565.7 | 68653.4 |  |  |  |

### Common Words

Following the pre-processing of the text data but before vectorisation we performed several analyses on the data. Once stop-words are removed we found the most common words in the data set.

A picture containing text, clipart

Description automatically generated

As we can see in figure X, the most common words across all categories are generic words with little meaning. These could be candidates to re-classify as stop-words as they are unlikely to differentiate between categories.

By analysing the most frequent occurrences of words in each category we can see if there are real differences between categorisations. Such processes could also identify addition project-specific stop-words which should be removed as part of the pre-processing procedure.

There is a marked difference in the top 10 most common words between each category . For example for the category rec.sport.hockey, the words “*team*”, “*game*” and “*player*” appear most frequently. In contrast the words “*drive*”, “*scsi*”, “*card*” and “*system*” appear most frequently for the comp.sys.ibm.pc.hardware category.

A picture containing text, clipart

Description automatically generated A picture containing text, clipart

Description automatically generated

### Bigrams

Bigrams are pairs of words which appear consecutively in the text. In some cases pairs of words which appear frequently together hold a specific meaning and could have the potential to improve a neural network model.

In this analysis bigrams were identified using *nltk bigram*. The most common bigram “maxaxax..” is a collection of letters found in every document of the *computer* category and therefore could be a significant factor when categorising documents.

Other common bigrams are more familiar combinations such as “dont know”, “law enforcement” and “san francisco”.

Chart

Description automatically generated

Pointwise mutual information (PMI) is a measure of association. **BigramAssocMeasures** from nltk (in conjunction with **BigramCollocationFinder**, a more sophisticated method of identifying bigrams) calculates the PMI, which for the first 20 bigrams gave a value of 20.30. This suggests there are no significant differences between these bigrams.

### Part of Speech (POS) Analysis

In the English language words can be classified according to their grammatical properties. After applying tokenisation we identified the number of nouns, verbs, adverbs, adjectives and other (that is any character that could not be classified into the aforementioned groups) using ***nltk pos\_tag***. For ease of comparability these results were scaled as a proportion of the total number of words in that category.

The POS composition (see figure x) shows each a similar proportion between each of the 20 categories, with the number of nouns accounting for roughly 50% of the total number of words.

A screenshot of a computer

Description automatically generated with medium confidence

## Model Implementation

## Tuning Hyperparameters

## Results and Error analysis

A description of the key evaluation metrics are as follows.

Precision is

Recall is

The F-score (or F1-score) is a measurement of the accuracy of a binary classifier built upon the precision and recall metrics already mentioned and is defined as follows;

whereby TP is the number of true positives classified, FP is the number of false positives and FN is the number of false negatives classified by the models.

### Confusion matrix

Evaluation of Models

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F1 score** |
| A |  |  |  |
| B |  |  |  |

Effectiveness by group for best performing model

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

## Conclusion

## References