AIT - Final Project
Using Advanced NLP Techniques to Categorize News Articles

Alan Gaugler
U885853@uni.canberra.edu.au

Faculty of Science and Technology
University of Canberra
ACT, Australia

Abstract

We live in a digital age, in which vast quantities of news and other content can be accessed and consumed online. Accurately categorizing all this content and recommending it to online users is a difficult challenge. In the past decade, Natural Language Processing (NLP) techniques have become necessary to classify online articles, based on their content.

This project utilizes the News Category Dataset from HuffPost [1]. It contains new headlines from approximately 210,000 articles that span a decade from 2012 to 2022. The project uses NLP algorithms to categorize the headlines into 42 distinct article categories. This midsemester report summarizes the work done thus far on the project. Extensive exploratory data analysis reveals insights into the data's structure. There is a notable class imbalance in this dataset that will be further investigated. The dataset has been processed and prepared for simpler machine learning models which serve as a baseline. This includes the implementation of TF-IDF vectorization. The two baseline models are Naïve Bayes and Logistic Regression, which have achieved classification accuracies of 52.5% and 60.5% respectively for the 42-class dataset.

The next stage of the project focused on further enhancing the data preparation and utilizing advanced deep learning models including Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) models. By merging some similar categories, the number of classes was reduced to 31, which resulted in a significant improvement in accuracy. The GRU and LSTM models were developed and tuned, and both resulted in improving the prediction accuracy over the baseline Logistic Regression model. The best accuracy was achieved by the GRU. The best overall classification accuracy was 69.3% which is a significant improvement over the baseline of 60.5%. This project has demonstrated that neural networks are better suited to NLP tasks than traditional models, but feature engineering and data preparation play an even more important role in improving classification accuracy.

1 Introduction

In the digital age of today, vast quantities of updated information and news are available to us constantly. For someone interested in reading the latest developments or researching past articles on a particular topic, the news article must first be categorized so that it can easily be identified as content that might be of interest and recommended to the user. With millions of

news articles available online, researching, categorizing, and filtering through such quantities is impractical for a human to do. Natural language processing (NLP) is required to categorize these articles.

NLP is a branch of Artificial Intelligence (AI) concerned with training machines to read, understand, interpret, and generate spoken and written human languages in a similar way that humans can [2]. In NLP, algorithms are developed to process large amounts of language data such as articles or documents, from which practical meaning and information can be retrieved.

One common and important application of NLP is article classification or categorization. This project is concerned with article classification. A news category dataset with over 200,000 article headlines and descriptions will be used in this project. The aim is to read and interpret the headlines and descriptions and categorize them into one of 42 topic categories. To achieve this, the project will be divided into several steps.

1. **Literature Review:** A literature review will be conducted to research existing methods and models used in NLP for article classification as well as possible challenges and limitations. The idea is to deploy a baseline first and then use more advanced techniques to find the optimal solution to this challenge.

2. **Exploratory Data Analysis (EDA):** The dataset will be investigated to determine what insights can be gained from it and to determine its quality and current suitability to be used for this NLP project. Data wrangling will be performed to make the dataset more suitable for the project.

3. **Data Preparation:** The text data will be cleaned, pre-processed and vectorized (word embedded) to prepare it for the machine learning stage.

4. **Establish a Baseline:** A baseline model will be established to demonstrate the suitability of this dataset in the field of article classification.

5. **Identification of Challenges, and Improvement in Data Preparation:** Issues faced will be addressed and further techniques will be investigated to improve the data preparation to improve the modelling performance.

6. **Deployment of More Advanced Models and Techniques:** The literature review has identified several advanced algorithms that can be used in topic modelling. This step and the previous will be used and refined testing various of these more advanced model architectures to improve the news article classification accuracy.

2 Literature Review

As was pointed out in the introduction, the categorization of vast amounts of online articles into distinct news categories requires the use of NLP to make this possible. To better understand the various methods and challenges in this important field of AI, an extensive literature review was carried out to establish which methods and models would be most suitable for the challenges of this particular project.

My interest in undertaking an NLP project was because of the importance of this field in the digital age and there are many practical applications with which it can be used. Additionally, diversification of projects is also important to gain experience in new fields. This

is my first NLP project. The first step in any data science project is to find a good dataset that will challenge me to learn new skills. I carried out an extensive search on various platforms. Various dataset repositories were searched including Kaggle, UCI Machine Learning Repository, GitHub and Google Dataset Search among several others. I found the chosen dataset to be of particular interest as it is extensive with over 200,000 entries and it is a complex classification problem with 42 categories. The dataset is described in more detail in the next section. The references section is provided at the end of this report, which provides references to the most interesting and useful articles that aided me in this project.

Understanding the processes and steps involved in an NLP project is fundamental to being able to complete the project successfully. Many research articles and papers written on medium.com were reviewed to gain a better understanding of the requirements. This series of articles series of NLP articles written by Fahrad Malik [3] is an excellent comprehensive guide to what is required in an entire NLP project. It was referred to often.

Text cleaning is a very important step in the NLP pipeline and through the literature review, I was able to determine that many methods are available to configure and evaluate [4], [5], [6], [7]. The NLTK (Natural Language ToolKit) and Gensim libraries were the main ones chosen for this task. They are among the most utilized and highest-rated libraries in NLP.

Word embedding and vectorization are important steps in the NLP pipeline, performed after the text cleaning. Through the literature review, I was able to determine that many methods are available to configure and evaluate. Certain methods are more suitable for certain models. [8], [9], [10] Some of the more common and better-rated techniques are Word2Vec, Term Frequency-Inverse Document Frequencies (Tf-idf) and on the more advanced models, further techniques such as GloVe (Global Vector for word representation) and BERT (Bidirectional Encoder Representations from Transformers) can be deployed.

There are several classification algorithms that can be used in the NLP pipeline. For the baseline, after extensive review, it was decided to use Naïve-Bayes and Logistic Regression mainly due to their simplicity and computational efficiency [11], [12], [13], [14]. More advanced models will be deployed to gain the best results possible. The literature review has found from several sources that among the best models to evaluate are Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Transformer models such as Bidirectional Encoder Representations from Transformers (BERT), RoBERTa or other pre-trained models [15], [16], [17], [18].

3 Dataset

The dataset for this project is the News Category Dataset [1], which has been uploaded on Kaggle. This dataset contains 209,527 news headlines from 2012 to 2022 from HuffPost, which is a very popular and well-known American news website. It contains news, satire, blogs, and more original content. It covers a large variety of topics including politics, business, entertainment, environment, technology, popular media, lifestyle, culture, comedy, and local news featuring columnists.

	link	headline	category	short_description	authors	date
0	https://www.huffpost.com/entry/covid-boosters-...	Over 4 Million Americans Roll Up Sleeves For O...	U.S. NEWS	Health experts said it is too early to predict...	Carla K. Johnson, AP	2022-09-23
1	https://www.huffpost.com/entry/american-airlin...	American Airlines Flyer Charged, Banned For Li...	U.S. NEWS	He was subdued by passengers and crew when he ...	Mary Papenfuss	2022-09-23
2	https://www.huffpost.com/entry/funniest-tweets...	23 Of The Funniest Tweets About Cats And Dogs ...	COMEDY	"Until you have a dog you don't understand wha...	Elyse Wanshel	2022-09-23
3	https://www.huffpost.com/entry/funniest-parent...	The Funniest Tweets From Parents This Week (Se...	PARENTING	"Accidentally put grown-up toothpaste on my to...	Caroline Bologna	2022-09-23
4	https://www.huffpost.com/entry/amy-cooper-lose...	Woman Who Called Cops On Black Bird-Watcher Lo...	U.S. NEWS	Amy Cooper accused investment firm Franklin Te...	Nina Golgowski	2022-09-22

Figure 1 – The Header of the News Category Dataset

Each record in the dataset consists of the following attributes:

- link: link to the original news article.
- headline: the headline of the news article.
- category: category in which the article was published.
- short description: Abstract of the news article.
- authors: list of authors who contributed to the article.
- date: publication date of the article.

There are 42 categories in the dataset making it a complex and challenging natural language processing classification project.

4 Exploratory Data Analysis

A comprehensive exploratory data analysis (EDA) was performed on this dataset. For a comprehensive overview of all the work performed in the EDA section, please refer to the attached Jupyter notebook, where all the work done has been described. In this report, the most important observations and actions taken that will affect the machine learning process will be described.

First, it had to be ensured that the dataset was valid and suitable for a project of natural language processing. It was confirmed that there are 209,527 news headlines in the dataset spanning 2012 to 2022. There are no missing values in the dataset. This will eliminate the need for data imputation.

It was confirmed that there are 42 article categories in the dataset, however, there is a very large class imbalance as is shown in Figure 2.

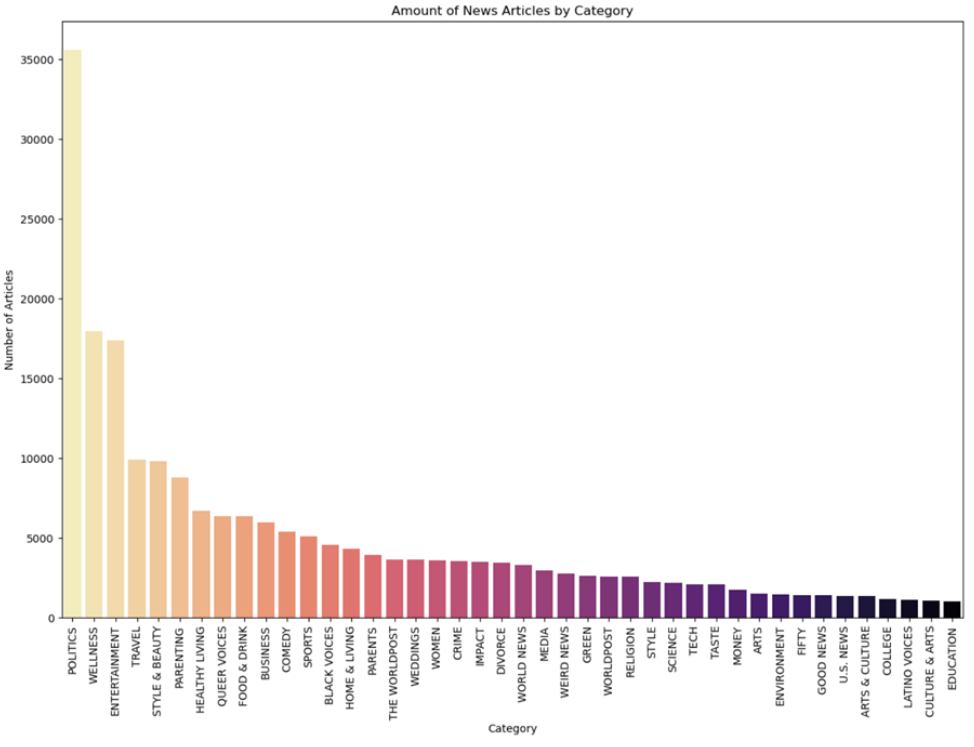

Figure 2 – The Amount of News Articles by Class

The nature of the Huffington Post is to cover political news more than any other type, hence why there is a large imbalance leaning towards this category. The next most common categories are Wellness, Entertainment, Travel, Style Beauty and Parenting. This large class imbalance will cause issues with detecting the minor classes. The class imbalance will be further investigated before the completion of the project. Some techniques to tackle this issue will be investigated such as SMOTE, oversampling or under-sampling.

For the two columns containing the text information that would be used in the NLP, some duplicate content was found. This could potentially bias the models for these categories.

```
|:  1  # Are there any duplicate headlines?
    2  df['headline'].value_counts()
```

```
|:  Sunday Roundup                                          90
   The 20 Funniest Tweets From Women This Week              80
   Weekly Roundup of eBay Vintage Clothing Finds (PHOTOS)   59
   Weekly Roundup of eBay Vintage Home Finds (PHOTOS)       54
   Watch The Top 9 YouTube Videos Of The Week               46
```

Figure 3 – Value Counts for the 'Headline' Feature

For 'headline', the maximum number of duplicate occurrences is 90 for the headline "Sunday Roundup'.

```
# Are there any duplicate short descriptions?
df['short_description'].value_counts()
```

```
                                                                                                                    19712
Welcome to the HuffPost Rise Morning Newsbrief, a short wrap-up of the news to help you start your day.              192
The stress and strain of constantly being connected can sometimes take your life -- and your well-being -- off course. GPS    125
Want more? Be sure to check out HuffPost Style on Twitter, Facebook, Tumblr, Pinterest and Instagram at @HuffPostStyle. -- Do    91
Do you have a home story idea or tip? Email us at homesubmissions@huffingtonpost.com. (PR pitches sent to this address will    75
                                                                                                                     ...
```

Figure 4 – Value Counts for the 'Short Description' Feature

For 'short description', the maximum number of duplicates is 192, as is seen above. To reduce the number of duplicates and to make it easier for the NLP, these two features were merged into one by concatenating them. This feature is called 'combined info'.

```
# Are there any duplicates in the combined info column?
df['combined_info'].value_counts()
```

```
Watch The Top 9 YouTube Videos Of The Week If you're looking to see the most popular YouTube videos of the week, look no fur
ther. Once again, we're bringing you the                                                                              46
The Funniest Tweets From Women This Week
33
The 20 Funniest Tweets From Women This Week The ladies of Twitter never fail to brighten our days with their brilliant — but
succinct — wisdom. Each week, HuffPost Women                                                                          30
Best Parenting Tweets: What Moms And Dads Said On Twitter This Week Kids may say the darndest things, but parents tweet abou
t them in the funniest ways. So each week, we round up the most hilarious                                             26
Funniest Parenting Tweets: What Moms And Dads Said On Twitter This Week Kids may say the darndest things, but parents tweet
about them in the funniest ways. So each week, we round up the most hilarious                                         23
```

Figure 5 – Value Counts for the 'Combined Info' Feature

This has resulted in the highest number of duplicates being reduced to just 46. 46 out of 209,527 is extremely low and will not bias the results significantly. These are articles that were actually published, so they will not be removed from this project.

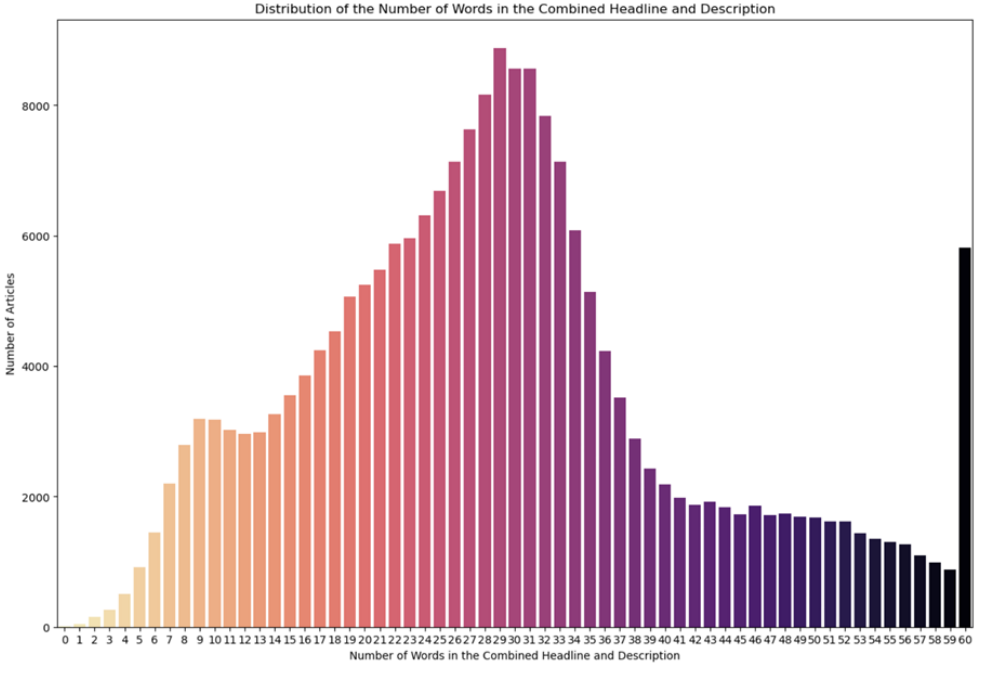

Figure 6 – Distribution of the Number of Words in the 'Combined Info' Feature

Five rows were found to not have any textual information at all (0 words) in the 'combined info' column, and 48 rows only had one word which was not very indicative of the category (Figure 7). With columns containing such few words, no significant information can be conveyed about what category they belong to. These rows were removed. This leaves 209,474 headlines.

category	short_description	authors	date	word_count_headline	word_count_description	combined_info	word_count_combined
COMEDY		Tom Kramer, ContributorWriter of the Wry	2016-06-26	1	0	"ManScraping"	1
COMEDY		Marcia Liss, Contributor(Almost) Famous Cartoo...	2016-05-29	1	0	Tire-d	1
TASTE		Dough Mamma, ContributorPrivate chef, culinary...	2016-05-29	1	0	Wafflewich	1
COMEDY		Marcia Liss, Contributor(Almost) Famous Cartoo...	2016-03-19	1	0	Hangman	1
COMEDY		Marcia Liss, Contributor(Almost) Famous Cartoo...	2016-01-10	1	0	Hugs	1
COMEDY		Marcia Liss, Contributor(Almost) Famous Cartoo...	2015-12-06	1	0	Memories	1
POLITICS		Gabriela Rivera-Morales, ContributorBlog Edito...	2015-12-06	1	0	IGNORE.	1

Figure 7 – A Sample of "Combined Info" Containing Just One Word

With the EDA completed, the next stage of the pipeline could commence which is the Data Preparation.

5 Data Preparation

Text Cleaning and Preprocessing

To prepare the textual data for natural language processing, the text requires several preprocessing steps to be carried out to transform it into a format that is suitable for the machine learning (ML) process to use with maximum benefit. The article headlines and descriptions were cleaned and converted into structured and numerical data for the ML modelling as follows:

1. **Text Cleaning:** This preprocessing step converted the text into lowercase, stripped and removed punctuation, effectively cleaning the text for further processing.
2. **Expand contractions:** Contracted words were converted into two words which make more sense. Additionally, the apostrophes were removed. An example of this is: "I'd" –> "I would". Many of these words will later be removed by the stop word removal step.
3. **Tokenization:** The entire body of text was partitioned into individual words and symbols. These are called tokens.
4. **Stop word removal:** This removed very common words that convey no meaning about the article such as "he", "she" or "on".

5. **Stemming:** This reduced words to their root form i.e., "shows", "showing" and "showed" will be reduced to "show".

6. **Lemmatization:** This also reduced words to their root form i.e., "better" and "best" will be reduced to "good". This is similar to stemming but the root words are more often real words used in English rather than just their stems.

7. **Bigrams and trigrams:** Many words often connected in sequence may have a different meaning and should be joined together such as "New" followed by "York" really conveys the meaning of a city called "New York" and they should be considered one word.

8. **Removal of unique words:** Unique words or words that occur only one time in the entire dataset will be removed. As these words are unique, they will not be encountered in the test set if they are in the training set and so will not convey any information in identifying the topic category. This will also reduce the "noise" in the dataset and speed up processing.

Text Vectorization
The data was then split into train and test sets and then text vectorization was applied. Machine learning models require numerical input rather than textual, so text needs to be converted into vectors. Converting text into numerical data is called 'vectorization' or 'embedding'. Three methods were evaluated:

Bag of words: This is a basic method that converts text to vectors. Each word in the corpus is given an index and the word's frequency is associated with it. There are no more complex structures for this method.

Term Frequency-Inverse Document Frequency: (tf-Idf). This is a more advanced method. Instead of just counting the number of words, tf-Idf also adjusts word values based on their occurrence frequency in all the headline rows, reducing the weight of the terms that appear more frequently in the headlines. This will emphasize words that are more unique to a certain category.

Word2vec: This is a more advanced neural network-based algorithm that learns word associations from a large corpus of text. Word2vec creates vectors of the words that are distributed numerical representations of word features – these word features could comprise of words that represent the context of the individual words present in our vocabulary. Word embeddings eventually help in establishing the association of a word with another similar meaning word through the created vectors. Credit: Analytics Vidhya [19].

6 Baseline Models

In using NLP for content classification, it is important to first establish that the dataset is suitable for the defined task and that baseline models can be successfully used to produce meaningful results, before progressing to more advanced models and techniques. For this dataset, two models have been used as a baseline. A Multinomial Naive-Bayes model and

a Logistic Regression model. They were chosen due to various reasons [13], [20] which include the following:

Simplicity and Efficiency: This is the main reason they were chosen for a baseline. Both models are relatively simple models that are well established, very easy to train and very efficient in computational speed and resources.

Good Accuracy: Both offer reasonably good accuracy. The Logistic Regressor offers good accuracy for many NLP datasets, in particular, if the dataset is linearly separable.

Multiple Classes: They work well with multiple classes such as this dataset.

The purpose of this section for the mid-semester report is to demonstrate that a baseline model can be used to produce an output on the chosen dataset. The models have been left with their default hyperparameter settings and were not tuned. For the final report, more advanced models will be evaluated and tuned to obtain maximum performance. Additionally, more feature engineering will be investigated and applied.

In the baseline, an extensive evaluation of various text processing and vectorization steps was carried out to determine which steps work best for this dataset. The table shown below summarises the various configurations that were tested.

Model	Classifier	Lemmatization Stemmming or Both	Contraction of Words	Remove Unique Words	Bigrams	Trigrams	Tfidf Vecotrizer or Other	Accuracy (%)	Comments
1	Logistic	Lemm	N	N	N	N	max_features=5000	58.30	
2	Multinomial NB	Lemm	N	N	N	N	max_features=5000	52.33	
3	Logistic	Lemm	Y	Y	Y	Y	max_features=5000	54.09	
4	Multinomial NB	Lemm	Y	Y	Y	Y	max_features=5000	48.38	
5	Logistic	Lemm	Y	N	N	N	max_features=5000	58.28	
6	Multinomial NB	Lemm	Y	N	N	N	max_features=5000	52.20	
7	Logistic	Lemm	N	Y	N	N	max_features=5000	58.32	
8	Multinomial NB	Lemm	N	Y	N	N	max_features=5000	52.33	
9	Logistic	Lemm	N	Y	Y	N	max_features=5000	55.30	
10	Multinomial NB	Lemm	N	Y	Y	N	max_features=5000	49.44	
11	Logistic	Lemm	N	Y	N	N	max_df=0.95, min_df=2	60.34	
12	Multinomial NB	Lemm	N	Y	N	N	max_df=0.95, min_df=2	44.56	
13	Logistic	Stem	N	Y	N	N	max_features=5000	58.83	
14	Multinomial NB	Stem	N	Y	N	N	max_features=5000	52.48	
15	Logistic	Stem	N	Y	N	N	max_features=20000	60.44	
16	Multinomial NB	Stem	N	Y	N	N	max_features=20000	47.65	
17	Logistic	Stem	N	Y	N	N	max_features=50000	60.51	This is the best Logistic Configuration
18	Multinomial NB	Stem	N	Y	N	N	max_features=50000	43.64	
19	Logistic	Stem	N	Y	N	N	max_df=0.95, min_df=2	60.38	
20	Multinomial NB	Stem	N	Y	N	N	max_df=0.95, min_df=2	45.06	
21	Logistic	Both	N	Y	N	N	max_features=5000	58.83	
22	Multinomial NB	Both	N	Y	N	N	max_features=5000	52.42	
23	Logistic	Both	N	Y	N	N	ngram_range=(1, 2)	NA	MUCH SLOWER!
24	Multinomial NB	Both	N	Y	N	N	ngram_range=(1, 2)	39.02	
25	Logistic	Both	N	Y	N	N	word2vec	52.64	
26	Multinomial NB	Both	N	Y	N	N	word2vec	NA	Cannot take negative values
27	Logistic	Stem	N	Y	N	N	bag of words	59.26	
28	Multinomial NB	Stem	N	Y	N	N	bag of words	57.81	This is the best NB configuration

Table 1 – Evaluation of Preprocessing and Vectorization Steps Applied to the Two Baseline Models

Looking at the evaluation table above, the logistic regression model consistently performed better than the Naïve-Bayes model. The best configuration was when stemming was applied, rather than lemmatization. This difference was, however, extremely slight. Removing the unique words slightly improved the performance. The performance degraded on this dataset when the contraction of words, bigrams and trigrams were used. These three steps were then omitted. The Tf-idf vectorizer with a large setting for max features outperformed the bag of words and word2vec vectorizers for these models. Word2vec will be used on the more advanced models in the next stage.

For the multinomial Naïve Bayes model, several of the minority classes had a precision, recall and an f1-score of 0.00. This indicates that these classes were never predicted. The best Naive-Bayes model achieved an accuracy of 57.8% using bag of words.

The logistic regression classifier always predicted some instances as belonging to every class, indicating it performs much better on this imbalanced dataset. However, the minority classes still tended to have a lower recall and f1 score than the majority classes. For example, "Fifty" compared to "Food Drink" in Figure 8. The best Logistic Regression model achieved an accuracy of 60.5% using a Tf-idf vectorizer. With an overall prediction accuracy of 60.51%, there is much room for improvement.

```
Accuracy: 60.51%

Classification Report:

                precision    recall  f1-score   support

          ARTS       0.43      0.24      0.31       302
ARTS & CULTURE       0.46      0.16      0.24       268
  BLACK VOICES       0.52      0.34      0.41       917
      BUSINESS       0.51      0.48      0.49      1198
       COLLEGE       0.51      0.35      0.41       228
        COMEDY       0.58      0.41      0.48      1077
         CRIME       0.57      0.56      0.57       712
CULTURE & ARTS       0.67      0.23      0.34       215
       DIVORCE       0.80      0.69      0.74       685
     EDUCATION       0.43      0.27      0.33       203
 ENTERTAINMENT       0.55      0.78      0.65      3472
   ENVIRONMENT       0.57      0.20      0.30       289
         FIFTY       0.49      0.14      0.21       279
  FOOD & DRINK       0.62      0.75      0.68      1268
```

Figure 8 – A Part of the Classification Report for the Best Performing Logistic Regression Model

Observing the categories of the news articles above, it appears that many categories are very similar and may be difficult to differentiate. Examples of this include:

“Arts”, “Arts Culture” and “Culture Arts”

“World News” and “Worldpost”,

“Style” and “Style Beauty”

In the remainder of the project, merging some of these categories will be investigated as it may well result in higher accuracy and also the removal of unnecessary subcategories.

7 Learning Outcomes

- The most important outcome of the project so far is that a baseline can be successfully run on this dataset.
- The data processing steps are very important for NLP projects and experimenting with various configurations can produce different results. As demonstrated in Table 1, various combinations were tried before determining which combinations of preprocessing and word embedding techniques worked optimally for this dataset. More techniques will be tried on the neural network models.
- Class Imbalance significantly affects the results of the minority classes. As was mentioned in the Modelling section, for the Naïve Bayes Model, some of the minority classes were never predicted and also for the Logistic Regressor, the recall on the minority classes is much lower than for the larger classes.
- The resulting best accuracy is only 60.51%. This is a challenging project and various more advanced models will be developed and evaluated to significantly improve on this score.
- Bigrams and trigrams led to performance degradation in these models. This is a surprising outcome to me as I would have expected an improvement. The bigrams were observed, and they seemed to be very logical such as people's names like Kevin Costner or cities like New York. Perhaps the increased dimensionality may have impacted the performance of these simpler models. Bigrams will be tried again on the more advanced models.

8 Scope of Continuing Work After the Midterm Report

The baseline work up to now has successfully established that this dataset is indeed good for Natural Language Processing of news article headlines and descriptions and that models are able to classify the articles into a large number of categories (42). The best classification accuracy produced in the baseline evaluations is 60.51%. This may seem reasonable for classifying 42 categories, but the scope of this project is to use more advanced models and feature engineering techniques to obtain as good an accuracy score as possible. Many approaches can be taken, some of which include:

Data Preprocessing and Feature Engineering
Further text cleaning: Although a comprehensive text cleaning procedure has been already implemented, there may be some further techniques that may enhance the modelling performance. There may be several keywords that should be exempt from preprocessing or modified prior to it such as U.S. to United States.

Merging of Categories: As mentioned in the Baseline Models section, many categories seem to be very similar and it may be beneficial to merge some of these categories, which by name seem very similar or indistinguishable to humans.

Word Embedding: is the collective name for feature learning techniques where words from the vocabulary are mapped to vectors of real numbers. These vectors capture the semantic context in which words appear within the corpus. [7]. Work will continue on Word2Vec, and Tf-idf on the newer models and further techniques can be applied such as GloVe (Global Vector for word representation) and BERT (Bidirectional Encoder Representations from Transformers) [8].

Class Imbalance: As has been noted, there is a significant class imbalance among the 42 classes. Some techniques could potentially be applied including SMOTE (Synthetic Minority Over-sampling Technique), oversampling or undersampling.

Modelling
Advanced Algorithms: Several neural network type architectures can be utilized in NLP. Various of these will be evaluated. These include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Transformer models such as Bidirectional Encoder Representations from Transformers (BERT).

Hyperparameter Tuning: In the baseline, the objective was to ensure that the dataset could be used as well as try to improve some text preprocessing and feature engineering techniques. The best-performing models will have their hyperparameters tuned through a grid search to obtain optimal performance.

Ensemble Models: Often ensemble models yield more accurate results. This is combining various of the best models which will 'vote' for the most likely class a record belongs to.

Additional Work

Since the midterm report, much more additional work has been carried out. The additional work will be described in the following sections.

9 Merging of Similar Categories

The original dataset consists of 42 categories. It can be observed that many of these categories are very similar or even essentially identical. A good example is the categories of 'CULTURE & ARTS' and 'ARTS & CULTURE'. They are virtually the same category, and if a human cannot make an accurate distinction of which category an article headline falls into, then an algorithm almost certainly will not be able to make a clear distinction either. All category labels were reviewed and several of the content of examples of these articles were also carefully reviewed. As there was very little to no clear distinction of several articles in many categories, it was decided to amalgamate or merge several of these categories. The list displayed below shows the original category on the left and the new category it was

relabelled to on the right:

'PARENTS': 'PARENTING'
'THE WORLDPOST': 'WORLD NEWS'
'WORLDPOST': 'WORLD NEWS'
'BUSINESS': 'BUSINESS & FINANCE'
'MONEY': 'BUSINESS & FINANCE'
'COLLEGE': 'EDUCATION'
'STYLE': 'STYLE & BEAUTY'
'GREEN': 'ENVIRONMENT'
'ARTS': 'ARTS & CULTURE'
'CULTURE & ARTS': 'ARTS & CULTURE'
'HEALTHY LIVING': 'WELLNESS'
'TASTE': 'FOOD & DRINK'

This resulted in the reduction of 11 categories, with a total category count down from 42 to 31 (Figure 9). Rerunning the best baseline configuration, the classification accuracy of the logistic regression model increased by over 6% from 60.51% to 66.76%. This is a considerable improvement and a good reduction of unnecessary categories.

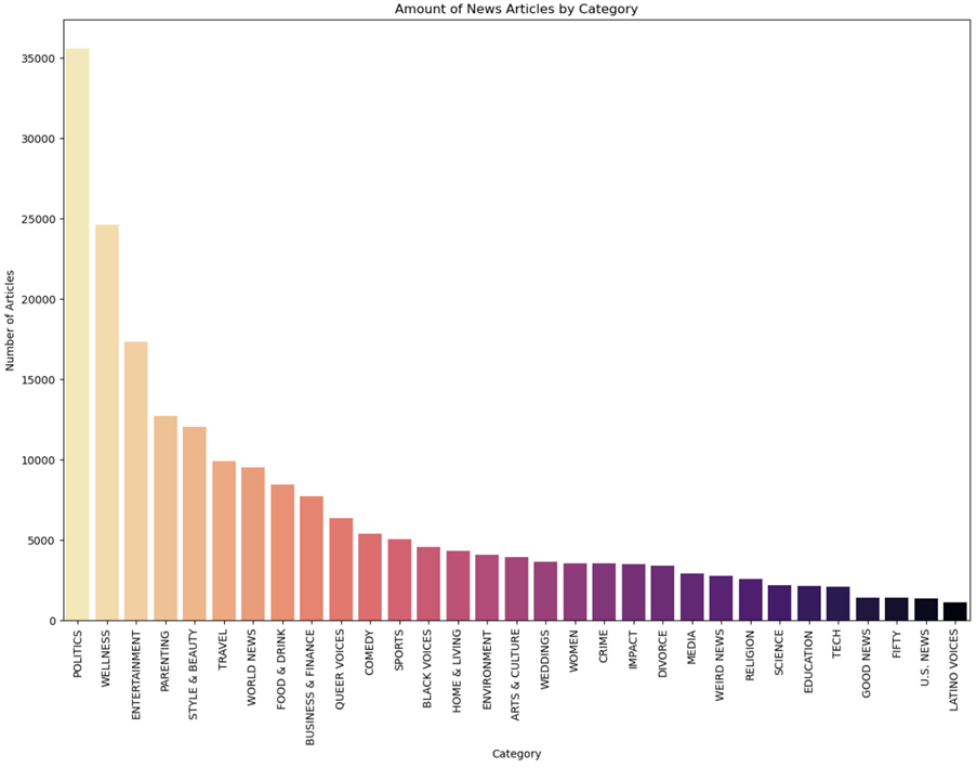

Figure 9 – Chart of The Number of News Articles by Class After Merging Categories

The summaries of the classification reports for the 42 and 31 categories modelling performance are shown in Figures 10 and 11. A comparison of the precision, recall and f1-scores

of the old categories in Figure 10 and the new categories in Figure 11 shows good improvement in the merged categories. For example, 'THE WORLDPOST', 'WORLDPOST' and 'WORLD NEWS' were all merged into the category of 'WORLD NEWS'. The precision, recall and f1-score in Figure 11 are all clearly better than for the three categories previously. 'PARENTS' and 'PARENTING' being merged also shows excellent improvement. All other categories show good improvement too.

This merger of similar categories leads to a simpler dataset. With fewer categories, the classification task is made easier as there is a reduced decision boundary for the models to learn.

Although still very significant, the class imbalance has been considerably reduced, making it somewhat more likely that the smaller classes will be predicted. In this dataset, it is not critical to predict the smaller classes, but any reduction in class imbalance is always good for the overall modelling accuracy.

Not only has the overall accuracy increased significantly. The macro average F1-score treats all classes equally and it has shown a significant increase from 46% to 54%. Additionally, the weighted average, which considers the weights (size) of each class has also increased by a similar margin from 58% to 65%. This measure gives more weight to the larger classes and suggests better performance in their metrics too. As most of these merged categories were smaller categories, this increase in accuracy is a significant improvement.

Considering the similarities of the merged categories and the overall improvement of the numerous accuracy metrics, it can be concluded that merging these categories was indeed a significant step in improving the overall accuracy of classifying these topics. These new categories will be used for further modelling.

Accuracy: 60.51%

Classification Report:

	precision	recall	f1-score	support
ARTS	0.43	0.24	0.31	302
ARTS & CULTURE	0.46	0.16	0.24	268
BLACK VOICES	0.52	0.34	0.41	917
BUSINESS	0.51	0.48	0.49	1198
COLLEGE	0.51	0.35	0.41	228
COMEDY	0.58	0.41	0.48	1077
CRIME	0.57	0.56	0.57	712
CULTURE & ARTS	0.67	0.23	0.34	215
DIVORCE	0.80	0.69	0.74	685
EDUCATION	0.43	0.27	0.33	203
ENTERTAINMENT	0.55	0.78	0.65	3472
ENVIRONMENT	0.57	0.20	0.30	289
FIFTY	0.49	0.14	0.21	279
FOOD & DRINK	0.62	0.75	0.68	1268
GOOD NEWS	0.48	0.14	0.22	279
GREEN	0.45	0.40	0.42	524
HEALTHY LIVING	0.42	0.21	0.28	1338
HOME & LIVING	0.75	0.72	0.74	864
IMPACT	0.47	0.29	0.36	697
LATINO VOICES	0.78	0.22	0.34	226
MEDIA	0.59	0.39	0.47	589
MONEY	0.55	0.37	0.44	351
PARENTING	0.52	0.64	0.57	1758
PARENTS	0.49	0.22	0.30	791
POLITICS	0.67	0.86	0.75	7120
QUEER VOICES	0.77	0.64	0.70	1269
RELIGION	0.61	0.44	0.52	515
SCIENCE	0.64	0.44	0.53	441
SPORTS	0.69	0.67	0.68	1015
STYLE	0.57	0.18	0.28	451
STYLE & BEAUTY	0.72	0.83	0.77	1963
TASTE	0.48	0.10	0.16	419
TECH	0.63	0.42	0.50	421
THE WORLDPOST	0.52	0.39	0.45	733
TRAVEL	0.66	0.79	0.72	1979
U.S. NEWS	0.34	0.04	0.08	275
WEDDINGS	0.79	0.78	0.79	731
WEIRD NEWS	0.41	0.25	0.31	555
WELLNESS	0.54	0.82	0.66	3589
WOMEN	0.47	0.32	0.38	714
WORLD NEWS	0.52	0.31	0.39	660
WORLDPOST	0.51	0.26	0.34	515
accuracy			0.61	41895
macro avg	0.57	0.42	0.46	41895
weighted avg	0.59	0.61	0.58	41895

Figure 10 – Classification Report of the Logistic Regression Model on the Original Dataset of 42 Categories

Accuracy: 66.76%

Classification Report:

	precision	recall	f1-score	support
ARTS & CULTURE	0.61	0.45	0.52	784
BLACK VOICES	0.55	0.34	0.42	917
BUSINESS & FINANCE	0.58	0.54	0.56	1549
COMEDY	0.61	0.39	0.48	1077
CRIME	0.58	0.53	0.55	712
DIVORCE	0.84	0.65	0.73	685
EDUCATION	0.51	0.37	0.43	431
ENTERTAINMENT	0.59	0.76	0.67	3472
ENVIRONMENT	0.53	0.44	0.48	813
FIFTY	0.46	0.14	0.21	279
FOOD & DRINK	0.75	0.78	0.76	1687
GOOD NEWS	0.45	0.10	0.16	279
HOME & LIVING	0.78	0.70	0.74	864
IMPACT	0.51	0.25	0.34	697
LATINO VOICES	0.82	0.27	0.40	226
MEDIA	0.63	0.37	0.46	589
PARENTING	0.65	0.72	0.68	2549
POLITICS	0.70	0.84	0.76	7120
QUEER VOICES	0.79	0.62	0.69	1269
RELIGION	0.59	0.41	0.49	515
SCIENCE	0.73	0.42	0.53	441
SPORTS	0.71	0.69	0.70	1015
STYLE & BEAUTY	0.77	0.81	0.79	2414
TECH	0.60	0.40	0.48	421
TRAVEL	0.72	0.76	0.74	1980
U.S. NEWS	0.47	0.03	0.06	275
WEDDINGS	0.81	0.72	0.76	731
WEIRD NEWS	0.47	0.22	0.30	555
WELLNESS	0.64	0.84	0.73	4927
WOMEN	0.49	0.28	0.36	714
WORLD NEWS	0.69	0.68	0.69	1908
accuracy			0.67	41895
macro avg	0.63	0.50	0.54	41895
weighted avg	0.66	0.67	0.65	41895

Figure 11 – Classification Report of the Logistic Regression Model on the Modified Dataset of 31 Categories

There is still a large imbalance between the majority and the minority classes, which will still be a challenge, however, this will not be to the same extent as before.

10 Undersampling

After merging the similar categories, there is still a noticeable class imbalance (Figure 9). This is confirmed by the numbers in Figure 10. Class imbalance is bad because it is an environment in which not all classes are treated equally by the algorithm. This will lead to bias towards the majority classes as the algorithm will have seen more samples of the larger classes and will have learned their characteristics better, tending to choose them more when predicting the classes. If a model predicts the majority classes more often, it will have a

higher overall accuracy, but this does not mean the model is better. Precision, recall and the F1-score are better estimates for gauging the classification accuracy when the dataset is highly imbalanced such as this one. Balancing the classes may bring improved results to the dataset in particular predicting the minority classes. Although predicting the minority classes in this dataset is not critical as they are not particularly important like cancer detection or credit card fraud detection, it is always good practice to explore all options to try to improve the modelling accuracy.

I investigated the feasibility of various methods to balance the classes [21], [22], [23], [24], [25]. I decided to try to undersample the majority classes because: The most common methods are:

Oversampling the Minority Classes: This process increases the number of samples of the minority classes by either duplicating them or fabricating them with techniques such as SMOTE for NLP (Synthetic Minority Over-sampling TEchnique). This is often regarded as the best method however, although the vectorization captures the semantic meaning of the sentences, some of them may sound strange in English. The major disadvantage of oversampling for this project is however the very large increase in the dataset that will be required to implement oversampling properly. With limited time the large increases required in processing time make it unfeasible to explore this option completely.

Undersampling the Majority Classes: The dataset is already very large with almost 210,000 news articles, totalling 85 megabytes in size. This has led to significant training times for many models, in particular the pre-trained models like BERT where the training time is just impractical. (The training was started, and it estimated 530 hours for one epoch on my machine.) To make significant savings in processing time, undersampling was tried as the only viable alternative.

A simple script using Scikit-learn was written to drastically reduce the size of the majority classes to be more in line with the minority classes. The dataset was reduced from 209,527 articles down to 50,000 articles in an attempt to balance the dataset plus save a significant amount of processing time. The top 27 categories were reduced to a size of 1656 entries. The most drastic of these was politics which originally had 35,602 articles and is given far too much weight.

Four categories were below this number, with ‘Latino Voices’ being the smallest at 1130 articles. This is very close to being an evenly balanced dataset. Figure 12 shows the articles by class before and after undersampling.

POLITICS	35598	BUSINESS & FINANCE	1656
WELLNESS	24633	COMEDY	1656
ENTERTAINMENT	17360	WEDDINGS	1656
PARENTING	12746	IMPACT	1656
STYLE & BEAUTY	12068	RELIGION	1656
TRAVEL	9897	TRAVEL	1656
WORLD NEWS	9540	QUEER VOICES	1656
FOOD & DRINK	8435	WELLNESS	1656
BUSINESS & FINANCE	7745	WOMEN	1656
QUEER VOICES	6346	HOME & LIVING	1656
COMEDY	5384	BLACK VOICES	1656
SPORTS	5075	MEDIA	1656
BLACK VOICES	4583	STYLE & BEAUTY	1656
HOME & LIVING	4320	WORLD NEWS	1656
ENVIRONMENT	4066	DIVORCE	1656
ARTS & CULTURE	3922	FOOD & DRINK	1656
WEDDINGS	3653	CRIME	1656
WOMEN	3570	PARENTING	1656
CRIME	3562	SCIENCE	1656
IMPACT	3483	EDUCATION	1656
DIVORCE	3426	ENTERTAINMENT	1656
MEDIA	2943	WEIRD NEWS	1656
WEIRD NEWS	2776	ENVIRONMENT	1656
RELIGION	2576	SPORTS	1656
SCIENCE	2206	POLITICS	1656
EDUCATION	2157	ARTS & CULTURE	1656
TECH	2104	TECH	1656
GOOD NEWS	1397	GOOD NEWS	1397
FIFTY	1396	FIFTY	1396
U.S. NEWS	1377	U.S. NEWS	1377
LATINO VOICES	1130	LATINO VOICES	1130
Name: category_red, dtype: int64		Name: category_red, dtype: int64	

Figure 12 – The Number of News Articles by Class After Merging Categories and After Undersampling

One run was made on this new undersampled and balanced dataset. The classification report is shown in Figure 13.

Accuracy: 56.98%

Classification Report:

	precision	recall	f1-score	support
ARTS & CULTURE	0.51	0.54	0.53	331
BLACK VOICES	0.52	0.46	0.49	331
BUSINESS & FINANCE	0.48	0.49	0.49	332
COMEDY	0.55	0.46	0.50	332
CRIME	0.53	0.61	0.57	331
DIVORCE	0.79	0.77	0.78	331
EDUCATION	0.64	0.64	0.64	332
ENTERTAINMENT	0.42	0.47	0.44	331
ENVIRONMENT	0.54	0.52	0.53	331
FIFTY	0.47	0.39	0.42	279
FOOD & DRINK	0.66	0.77	0.71	331
GOOD NEWS	0.51	0.40	0.45	280
HOME & LIVING	0.68	0.76	0.72	331
IMPACT	0.43	0.40	0.42	332
LATINO VOICES	0.80	0.44	0.57	226
MEDIA	0.63	0.63	0.63	331
PARENTING	0.46	0.57	0.51	331
POLITICS	0.50	0.56	0.53	331
QUEER VOICES	0.77	0.63	0.69	331
RELIGION	0.69	0.66	0.67	331
SCIENCE	0.62	0.61	0.61	332
SPORTS	0.64	0.69	0.67	331
STYLE & BEAUTY	0.63	0.67	0.65	331
TECH	0.61	0.64	0.62	331
TRAVEL	0.61	0.68	0.64	331
U.S. NEWS	0.45	0.29	0.35	276
WEDDINGS	0.80	0.79	0.79	331
WEIRD NEWS	0.38	0.40	0.39	331
WELLNESS	0.43	0.54	0.48	331
WOMEN	0.46	0.43	0.44	331
WORLD NEWS	0.60	0.62	0.61	331
accuracy			0.57	10003
macro avg	0.57	0.57	0.57	10003
weighted avg	0.57	0.57	0.57	10003

Figure 13 – Classification Report of the Logistic Regression Model After Undersampling

Comparing the overall accuracy of the full dataset (Figure 11) and the under-sampled dataset (Figure 13), there are some notable differences.

1. **Overall Accuracy:** This is the most notable difference. There is a drastic reduction in accuracy from 66.76% down to 56.98%. This is most likely due to the model being trained on less data (words and sentences), it has not captured nearly the same amount of information for each category, resulting in much poorer prediction accuracy on the test set.

2. **Precision, Recall and F1 Score:** The weighted average for these three metrics has dropped also from 66% down to 57%, however, the changes are more subtle for the macro average which considers all classes equal. The precision has dropped from 63%

to 57%, but the recall has risen from 50% to 57% and the F1-score has risen slightly from 54% to 57%. This indicates that for the smaller classes, their consideration in the classification and their accuracy have increased, which is beneficial for them, but this has come at the cost of a large reduction in the overall accuracy.

Balancing the dataset by undersampling the majority classes clearly did not result in a favourable outcome. This was not implemented. There were improvements in predicting the minority classes, but this has drastically affected the overall accuracy. Clearly, the larger amount of training data is very beneficial to a dataset like this, in which there is a large number of additional vocabulary, semantics and variability for the models to learn from. As there are no critical categories, and accurately classifying any particular category is no more important than any other, it can be concluded that utilising the full dataset is clearly more advantageous to build an overall more accurate model than having a fairly balanced dataset. As mentioned, SMOTE is a viable option worthy of exploration, but due to the much larger processing time required, it is not an option that can be explored.

11 Deep Learning

The Naïve Bayes and Logistic Regression models provided a great baseline from which to establish this project, but the real aim is to use more advanced techniques to maximize the overall accuracy of this classification project. Further models and pre-embedding techniques were investigated and evaluated. Extensive combinations of hyperparameters were also evaluated and tuned. The results of all the evaluations are summarized in Table 2. As can be seen, several combinations of text preparation, neural network models and hyperparameters were evaluated. They will be concisely summarized in the following subsections. Keras was used to build the GRU and LSTM models. Using Keras, the ModelCheckpoint function was used to save the model configuration with the lowest validation loss after each epoch. This would ensure that the best model would be implemented when running predictions, thus avoiding overtraining. It was verified that on the independent test set that the prediction accuracy was very similar to or even slightly higher than on the validation set.

Encoding, Tokenization and Sequence Padding

Before applying deep learning models to the pre-processed text data, further preprocessing steps must be applied. The target categories are encoded into integers, as neural networks require numbers. One-hot encoding is then applied to these integers, creating a large binary matrix of classes.

The text data from the ‘combined_info’ column is then converted into tokens, represented by numbers. Effectively, a unique integer is tagged onto each unique word. These tokens are ranked starting from the most frequently occurring words in descending order.

The text entries will have various lengths. Padding will fill the shorter entries with zeros up to the maximum length so that all sequences are of equal length. The same input dimensions are required for all text data in the ML process.

GRU – Gated Recurrent Unit

This is a type of recurrent neural network (RNN) with a simpler architecture than LSTMs, making them more time-efficient in training. GRUs are designed to handle sequential data

such as time series, audio and text data by permitting information to be selectively remembered and forgotten over time. [26], [28] This flow of textual information is controlled by reset and update gates. The reset gate determines how much of the previous state should be forgotten, while the update gate determines how much of the new state should be remembered. This allows the GRU network to selectively update its internal state based on the input sequence [33]. GRUs tend to be better suited to short-term dependencies because they have a simpler architecture.

The architecture designed for this project is effective. It consists of:

Embedding Layer: This turns indices into vectors in the first layer of the model. Each word is transformed into a vector of high-dimensional space. A higher setting of this parameter of around 1000 worked better than smaller values. Going beyond 1000 there were diminishing returns with extended processing time.

GRU Layer: A simple architecture with 128 neurons, this worked well and adding more neurons tended to make a negligible difference. Making the GRU layer bi-directional processes the word sequence in a forward and reverse direction, adding past context to future context. This resulted in a substantial increase in overall accuracy up from approximately 66.5% to 67.5% with little increase in processing time.

Dense Layer: This fully connected layer is the final layer with the outputs numbering the number of classes. Softmax activation is used for multi-class classification.

Compile: Categorical cross entropy is the loss function used for multi-class classification. Adam and RMSProp optimizers were tested. RSMProp proved to be slightly more accurate in this case.

LSTM – Long Short-Term Memory
LSTMs are another type of RNNs that have a more complex design than GRUs [29]. It deals much better with the problem of long-term dependencies of simpler RNNs in which [30] they predict better at recent information but cannot predict text stored in long-term memory. LSTMs have three gating mechanisms that control the flow of text information through the network: the input gate, the forget gate, and the output gate [33]. These gates allow the LSTM network to selectively remember or forget information from the input sequence, which makes it more effective for long-term dependencies.

The architecture of the LSTM model for this project contains the following:

Embedding Layer: This turns indices into vectors in the first layer of the model. Each word is transformed into a vector of high-dimensional space. A higher setting of this parameter of around 1000 worked better than smaller values. Going beyond 1000 there were diminishing returns with extended processing time.

LSTM Layer: This can capture long-term dependencies better than a GRU and is more complex. Setting this layer bi-directional enables processing the word sequence in a forward and reverse direction, adding past context to future context. This resulted in a substantial increase in overall accuracy up from approximately 66.5% to 68.5% with little increase in

processing time. It was optimized to consist of 128 units (neurons) which worked best. Dropout has been disabled. Early stopping is implemented, and the best model checkpoint was saved.

Dense Layer: Originally consisted of half the neurons of the previous LSTM layer which is a good design convention. The ReLU activation function is the logical choice due to its efficiency. However it was found that a single layer LSTM worked better so this layer was removed.

Output Dense Layer: This fully connected layer is the final layer with the outputs numbering the number of classes. Softmax activation is used for multi-class classification.

Compile: Categorical cross entropy is the loss function used for multi-class classification. Adam and RMSProp optimizers were tested. RSMProp proved to be slightly more accurate in this case.

GloVe - Global Vectors for Word Representation
GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space [27]. To put is simply, GloVe places words into a vector space where similar words are clustered together and dissimilar words are not. It was developed by data scientists at the University of Stanford.

GloVe embeddings were incorporated into the GRU and LSTM models, to attempt that all the models have a more informed starting point of semantic relationships of the text. Table 2 shows that the results were not encouraging and were slightly worse than not using it. It was an option that was explored but did not produce a favourable outcome for this dataset.

Word2Vec - Global Vectors for Word Representation
This is a more advanced neural network-based algorithm that learns word associations from a large corpus of text. It was originally developed at Google in 2013. Word2vec creates vectors of the words that are distributed numerical representations of word features. These word features could comprise of words that represent the context of the individual words present in our vocabulary [31]. Word embeddings eventually help in establishing the association of a word with another similar meaning word through the created vectors.

Similar to GloVe, Word2Vec was tested on the GRU model but the results demonstrated that the prediction accuracy was down from around 68% to 65%. It was not explored further.

Model	Classifier	Units / Neurons	Tokenizer / Word Embedder	num_words	oov_token	Max Length	learning rate	embedding dimensions	Optimizer	dropout/ recurrent dropout	Test Accuracy (%)	Comments
1	GRU	32	Tokenizer	all	None	max	0.001	100	Adam	0.2	66.5	New words in test set ignored. great first run
2	LSTM	128/64	Tokenizer	all	None	max	0.001	100	Adam	0.2	16.99	New words in test set ignored.This led to only predicting the majority class
3	GRU	32	Tokenizer	15000	<UNK>	90th percentile	0.001	100	Adam	0.2	66.25	<UNK> instructs the tokenizer to include a specific token in its vocabulary for all out-of-vocabulary words.
4	LSTM	128/64	Tokenizer	15000	<UNK>	90th percentile	0.001	100	Adam	0.2	64.51	predicting more than 1 class
5	GRU	32	Tokenizer	20000	<UNK>	90th percentile	0.001	100	Adam	0.2	67.29	
6	LSTM	128/64	Tokenizer	20000	<UNK>	90th percentile	0.0001	100	Adam	0.2	64.52	
7	GRU	32	Tokenizer	10000	<UNK>	90th percentile	0.001	100	Adam	0.2	66.42	
8	LSTM	128/64	Tokenizer	10000	<UNK>	90th percentile	0.0001	100	Adam	0.2	64.51	
9	GRU	32	Tokenizer	all words	<UNK>	90th percentile	0.001	100	Adam	0.2	66.19	
10	LSTM	128/64	Tokenizer	all words	<UNK>	90th percentile	0.0001	100	Adam	0.2	66.57	
11	GRU	32	Tokenizer	all words	blank	90th percentile	0.001	100	Adam	0.2	66.19	
12	GRU	32	Tokenizer	all words	<UNK>	95th percentile	0.0005	100	Adam	0.2	65.17	
13	GRU	32	Tokenizer	all words	<UNK>	80th percentile	0.0005	100	Adam	0.2	64.65	
14	LSTM	128/64	Tokenizer	all words	<UNK>	80th percentile	0.0005	100	Adam	0.2	64.28	
15	GRU	32	Tokenizer/ GloVe	all words	<UNK>	80th percentile	0.0005	100	Adam	0.2	62.11	
16	GRU	32	Tokenizer/ GloVe	all words	<UNK>	90th percentile	0.0005	300	Adam	0	62.25	
17	GRU	32	Tokenizer	all words	<UNK>	90th percentile	0.0005	300	Adam	0	63.98	
18	LSTM	128/64	Tokenizer	all words	<UNK>	90th percentile	0.0005	300	Adam	0	65.85	
19	LSTM	128/64	Tokenizer/ GloVe	all words	<UNK>	90th percentile	0.0005	300	Adam	0	62.35	
20	GRU Bi-directional	32	Tokenizer	15000	<UNK>	90th percentile	0.001	100	Adam	0.2	67.31	
21	GRU Bi-directional	32	Tokenizer	all words	<UNK>	90th percentile	0.001	100	Adam	0.2	67.49	
22	LSTM - Bi-directional	128/64	Tokenizer	all words	<UNK>	90th percentile	0.001	100	Adam	0.2	66.30	
23	GRU Bi-directional	32	Tokenizer	all words	<UNK>	90th percentile	0.001	300	Adam	0.2	67.70	slower than 100
24	GRU Bi-directional	32	Tokenizer	all words	<UNK>	90th percentile	0.001	300	Adam	0	67.76	slower than 100
25	GRU Bi-directional	32	Tokenizer	all words	<UNK>	90th percentile	0.001	300	RMSProp	0	67.85	slower than 100
26	GRU Bi-directional	32	Tokenizer	all words	<UNK>	90th percentile	0.001	500	RMSProp	0	68.05	much slower than 100
27	GRU Bi-directional	32	Tokenizer	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	68.37	much slower than 500
28	GRU Bi-directional	32	Tokenizer	all words	<UNK>	90th percentile	0.001	3000	RMSProp	0	68.36	much slower than 1000, 4 mins per epoch to over 10 mins
29	GRU Bi-directional	64	Tokenizer	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	68.75	much slower than 500
30	GRU Bi-directional	128	Tokenizer	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	69.16	much slower than 500
31	GRU Bi-directional	32	Tokenizer/ GloVe	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	62.77	GloVe decreases performance
32	LSTM - Bi-directional	128/64	Tokenizer	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	67.68	GloVe decreases performance
33	LSTM - Bi-directional	128	Tokenizer/ GloVe	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	63.29	much slower than 500
34	GRU Bi-directional	128/64	Tokenizer/ GloVe	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	68.03	much slower than 500
35	GRU Bi-directional	256	Tokenizer/ GloVe	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	68.88	much slower than 500
36	GRU Bi-directional	128	Tokenizer/ GloVe	all words	<UNK>	90th percentile	0.001	1000	Adam	0	68.29	not as good as RMSProp
37	LSTM - Bi-directional	128	Tokenizer	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	68.64	The best LSTM, but GRU is better
38	GRU Bi-directional	128	Word2Vec	all words	<UNK>	90th percentile	0.001	1000	RMSProp	0	65.18	fast but less accurate
39	GRU Bi-directional	128	Tokenizer	all words	<UNK>	99th percentile	0.001	1000	RMSProp	0	69.31	much slower than 500
40	GRU Bi-directional	128	Tokenizer	all words	<UNK>	max	0.001	1000	RMSProp	0	69.32	much slower than 99th percentile
41	LSTM Bi-directional	128	Tokenizer	all words	<UNK>	max	0.001	1000	RMSProp	0	69.16	much slower than 99th percentile

Table 2 – Evaluation of Vectorization Steps and Deep Learning Models

Summary of Neural Network Testing

Both the GRU and the LSTM trained very fast. In most cases they reached minimum validation loss on only the second epoch, occasionally on the third. Afterwards they started to overtrain. Model checkpoint ensured the best trained and most accurate model of all epochs was used.

Several combinations of hyperparameter settings on the two models were combined with variations of the text embedding techniques (Table 2). For this configuration GloVe and Word2Vec text embedding slightly degraded the performance, so their testing was limited. The standard Encoding, Tokenization and Sequence Padding technique proved to work best for this project. The best configuration included an embedding dimension of 1000 words, anything longer increased the processing time with no significant improvement. The maximum length of all the sequences (the longest article description) yielded the best results. Truncating this decreased processing time but the results degraded slightly.

The best model configuration was a simple architecture, bi-directional GRU with a just single layer of 128 units (neurons). Figure 14 shows the best model configuration.

```
# Initialize the model structure for a GRU
gru1 = Sequential()

# Add an embedding layer
gru1.add(Embedding(vocab_size, embedding_dim, input_length=max_length))

# Add the GRU layer. Adding a bi-directional wrapper improved the performance
# Dropout and recurrent dropout worked best at 0.
gru1.add(Bidirectional(GRU(units=128, dropout=0, recurrent_dropout=0)))

# Add an additional Dense layer with ReLU activation - This did not improve performance
#gru1.add(Dense(64, activation='relu'))

# Final layer with 'softmax' for multi-class classification
gru1.add(Dense(number_of_categories, activation='softmax'))

# Compile the model. RMSProp worked better than Adam
gru1.compile(loss='categorical_crossentropy', optimizer=rmsprop_optimizer, metrics=['accuracy'])
```

```
# Display the model's summary
gru1.summary()
```

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_7 (Embedding)     (None, 142, 1000)         36338000

 bidirectional_6 (Bidirectio  (None, 256)              867840
 nal)

 dense_6 (Dense)             (None, 31)                7967

=================================================================
Total params: 37,213,807
Trainable params: 37,213,807
Non-trainable params: 0
```

Figure 14 – Architecture of the Best GRU Model

The best accuracy achieved was 69.32% using the GRU model. Figure 15 shows the classification report of the best model and Figure 16 shows the confusion matrix. Despite

the more complex design of the LSTM model, the GRU model usually performed better by about 1% on overall accuracy and in the other accuracy metrics too. One reason for this is that the 'combined_info' feature which is a merger of the article 'headline' and 'short description', contains an average of 29 words. This is a relatively short sequence of words. LSTMs are designed to handle far more complex sequences so it is highly possible that the GRU is capturing the text information more accurately.

Another likely reason is that there are still further architectural designs and hyperparameter setting combinations that can be explored further and it is very possible that with the right combination, the LSTM will perform better. More time and processing power is required to investigate further.

With a solid preprocessing technique optimized in the baseline, it proved to be a challenge at first to improve upon the Logistic Regression baseline. With extensive hyperparameter tuning, both RNNs outperformed the baseline logistic regressor.

RNNs have several advantages over traditional models in NLP. They are specifically designed to work with sequential data including text. They are better at identifying sequential patterns such as word order [32]. Traditional models such as Naïve-Bayes and Logistic Regressors take into account the words only and not their order or sequences. As was demonstrated in the project, RNNs are designed to process input sequences in both forward and backward directions. This allows the network to capture both past and future context, which is very advantageous in natural language processing tasks. RNNs have increased memory, allowing them to maintain certain information from previous sentences [30]. Through this they can gain a better understanding of the context. Logistic regression and other traditional models do not have this ability.

```
Accuracy: 69.32%

Classification Report:

                    precision    recall  f1-score   support

     ARTS & CULTURE       0.60      0.49      0.54       784
       BLACK VOICES       0.56      0.34      0.42       917
BUSINESS & FINANCE       0.57      0.59      0.58      1549
             COMEDY       0.54      0.49      0.51      1077
              CRIME       0.60      0.53      0.56       712
            DIVORCE       0.84      0.74      0.79       685
          EDUCATION       0.54      0.44      0.49       431
      ENTERTAINMENT       0.67      0.75      0.71      3472
        ENVIRONMENT       0.47      0.59      0.52       813
              FIFTY       0.49      0.20      0.28       279
       FOOD & DRINK       0.77      0.80      0.79      1687
          GOOD NEWS       0.39      0.19      0.26       279
      HOME & LIVING       0.86      0.70      0.77       864
             IMPACT       0.45      0.28      0.34       697
      LATINO VOICES       0.70      0.40      0.51       226
              MEDIA       0.60      0.38      0.47       589
          PARENTING       0.68      0.78      0.73      2549
           POLITICS       0.75      0.84      0.79      7120
       QUEER VOICES       0.77      0.67      0.72      1269
           RELIGION       0.59      0.54      0.56       515
            SCIENCE       0.69      0.45      0.54       441
             SPORTS       0.70      0.72      0.71      1015
     STYLE & BEAUTY       0.83      0.82      0.82      2414
               TECH       0.58      0.43      0.50       421
             TRAVEL       0.75      0.77      0.76      1980
          U.S. NEWS       0.53      0.06      0.11       275
           WEDDINGS       0.79      0.80      0.79       731
         WEIRD NEWS       0.41      0.32      0.36       555
           WELLNESS       0.70      0.83      0.76      4927
              WOMEN       0.51      0.31      0.39       714
         WORLD NEWS       0.69      0.74      0.71      1908

           accuracy                           0.69     41895
          macro avg       0.63      0.55      0.57     41895
       weighted avg       0.69      0.69      0.68     41895
```

Figure 15 – Sample Output of True and Predicted Labels

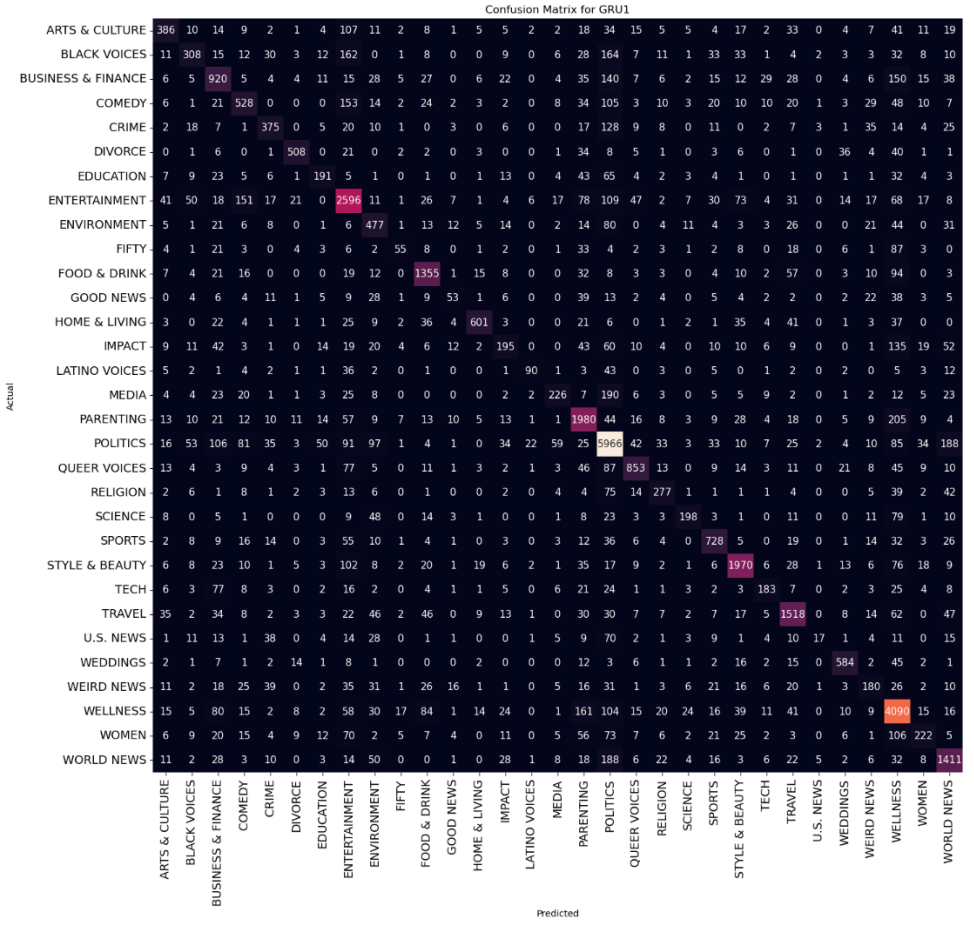

Figure 16 – Sample Output of True and Predicted Labels

12 Example of Predictions

Figure 17 shows a sample of the true and predicted labels and the 'processed_combined_info' column which was vectorized before machine learning. Stemming was chosen as a preprocessing step, so some words appear strange when reduced to their stems. In most cases in this example, the model predicted the correct category. In the mispredicted labels, reading them and making sense of the text, it can be easy to see why some labels were mispredicted.

Row 2: Crime was mispredicted as Weird News, likely due to a woman being bitten by a shark. This seems like something weird and not a crime.

Row 5: The true label of Weird News can easily be confused with Sports because of the words football, sport and touchdown.

Row 7: Also, Wellness was classified as Science and it can easily be seen why as the Nobel prize for medicine is mentioned.

Row 9: The example of World News being predicted as Entertainment is clearly a bad classification. This is likely due to the words 'Russian spy" often being a popular theme in movies.

These examples highlight the difficulty even for humans to categorize the brief headlines and descriptions correctly. Considering these examples, I would consider a classification accuracy of over 69% to be a good result. Undoubtedly reading the full article would have helped enormously in providing more accurate context, which would result in higher classification accuracy.

processed_combined_info	true_labels	predicted_labels
elder independ establish common ground whether care age parent neighbor apart build help home health aid connect ill spous look establish common ground drive forward success caregiv relationship	WELLNESS	WELLNESS
florida woman bitten shark inner tube	CRIME	WEIRD NEWS
sandal revolut may look go heck thing precis fashion commun put head togeth give shoe name reader like present newest hybrid shoe luxuri flatform	STYLE & BEAUTY	STYLE & BEAUTY
13 reason goe beyond tape haunt season 2 trailer new season debut netflix may 18	ENTERTAINMENT	ENTERTAINMENT
fowl bizarr bowl footbal next big sport mich ap detroit area entrepreneur believ score touchdown new busi idea thrown	WEIRD NEWS	SPORTS
st regi princevil pauper vill st regi love horizont ivori tower isol tip princevil leav kauai alway want	TRAVEL	TRAVEL
nobel prize medicin jame rothman randi thoma jointli win prize nobel committe said research traffic transport system cell help scientist understand	WELLNESS	SCIENCE
start viral pay homag youtub meme video parodi billi joel start fire go stuck head day	COMEDY	COMEDY
poison daughter russian spi releas hospit end treatment mark signific mileston	WORLD NEWS	ENTERTAINMENT
ed sheeran grammi nomin complet surpris expect year alreadi drunk found didnt expect year sheeran told huffington post z100	ENTERTAINMENT	ENTERTAINMENT
challeng patel poll show tighten race key new york primari patel take rep carolyn maloney jerri nadler democrat nomin manhattan congression district	POLITICS	POLITICS
trayvon martin case mean middl class black cri heard zimmerman verdict knew would tear countri apart racial line also knew middl class black person go call upon perspect white colleagu friend famili	BLACK VOICES	BLACK VOICES

Figure 17 – Sample Output of True and Predicted Labels

13 Conclusion

This was a comprehensive natural language processing project, in which several techniques of data preparation, data preprocessing and machine learning were evaluated and tested to produce a comprehensive report and gain a better understanding of all important aspects involved in natural language processing. Extensive Exploratory Data Analysis was carried out to gain a better understanding of the dataset. Thorough feature engineering and data preparation were performed including the many steps required in data preprocessing to prepare the text optimally for the machine learning process. A baseline was successfully created using Naïve-Bayes and Logistic Regression models. With these models in place, further optimization of the text preprocessing was performed in the baseline.

With the optimized preprocessing, the best baseline model was a Logistic Regressor which had an overall accuracy of 60.51%. Stage 2 of the project was to employ further feature engineering and modelling techniques to improve upon the baseline.

It was observed that of the 42 categories of news articles, several were very similar which would lead to common misclassifications. After careful consideration, several cat-

egories were merged, leaving 31 categories of news articles. This provided a significant improvement of the baseline, with the overall accuracy with the same logistic regression model improving to 66.45%. This large increase underscores the importance of good data preparation, which usually results in larger gains than implementing a more advanced model.

An attempt was made to balance the dataset by drastically reducing the majority classes by employing under-sampling techniques. Although the recall improved slightly, the overall accuracy fell substantially so it was concluded to not implement under-sampling on this NLP project. Too much valuable information had been lost in the dataset reduction.

Finally, a GRU and an LSTM model were built using Keras and various configurations of the architecture, hyperparameters and word embedding techniques were investigated. The word embedding techniques of GloVe and Word2Vec slightly degraded the performance of the two RNN models. A standard tokenization and sequence padding technique proved to work best for this configuration.

The best accuracy achieved was 69.32% using the GRU model. Despite the more complex design of the LSTM model, the GRU model consistently performed better by about 1% on overall accuracy and in the other accuracy metrics too. Additionally, there are still further architectural designs and hyperparameter setting combinations that can be explored further, possibly leading to better LSTM results.

Although my testing was extensive, it is certainly not 100% complete. Several more avenues could be explored given sufficient time.

This project proved to be very interesting and a great learning experience for me. There are still many areas of exploration left for this project to improve upon the results. Many of those mentioned in the "Scope of Continuing" Work section could not be carried out due to time constraints and lack of processing power. Hyperparameter tuning, further work with word embedding techniques and evaluation of more advanced pre-trained models such as BERT and DistilBERT and even ensemble methods are all areas worthy of further exploration. To summarize, an improvement from 60.5% to 69.3% is considerable and various methods were integrated together to achieve this solid gain.

References

[1] R. Misra, “News Category Dataset,” www.kaggle.com, May 2018. https://www.kaggle.com/datasets/rmisra/news-category-dataset

[2] L. Juliff, “Natural Language Processing,” School of Computing and Information Systems, Mar. 10, 2023. https://cis.unimelb.edu.au/research/artificial-intelligence/research/Natural-Langu

[3] F. Malik, “End To End Guide For NLP Project,” FinTechExplained, Aug. 02, 2019. https://medium.com/fintechexplained/end-to-end-guide-for-nlp-project-55e6765f63b5

[4] Sciforce, “Text Preprocessing for NLP and Machine Learning Tasks,” Medium, May 05, 2020. https://medium.com/sciforce/text-preprocessing-for-nlp-and-machine-learning-tasks-3e077aa4946e

[5] F. Malik, “NLP: Text Processing In Data Science Projects,” FinTechExplained, Jul. 30, 2019. https://medium.com/fintechexplained/nlp-text-processing-in-data-science-projects-f0830

[6] Gensim, "Gensim: topic modelling for humans," radimrehurek.com. https://radimrehurek.com/g

[7] S. Shenoy, "Elegant Text Pre-Processing with NLTK in sklearn Pipeline," Medium, Nov. 09, 2022. https://towardsdatascience.com/elegant-text-pre-processing-with-nltk-in-sklearn-pipeline-d6fe18b91eb8

[8] Turing, "Word embeddings in NLP: A Complete Guide," www.turing.com. https://www.turing.c on-word-embeddings-in-nlp

[9] J. S. Chawla, "Word Vectorization using GloVe," Analytics Vidhya, Jul. 06, 2020. https://medium.com/analytics-vidhya/word-vectorization-using-glove-76919685ee0b

[10] M. D. Pietro, "Text Classification with NLP: Tf-Idf vs Word2Vec vs BERT," Medium, Jul. 18, 2020. https://towardsdatascience.com/text-classification-with-nlp-tf-idf-vs-word2vec-vs-bert-41ff868d1794

[11] S. Li, "Multi-Class Text Classification Model Comparison and Selection," Towards Data Science, Sep. 25, 2018. https://towardsdatascience.com/multi-class-text-classification-model-comparison-and-selection-5eb066197568

[12] A. Makarov, "Best Architecture for Your Text Classification Task: Benchmarking Your Options," KDnuggets, Apr. 10, 2023. https://www.kdnuggets.com/2023/04/best-architecture-text-classification-task-benchmarking-options.html

[13] A. V. Ratz, "Multinomial Nave Bayes' For Documents Classification and Natural Language Processing (NLP)," Medium, Apr. 08, 2022. https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848

[14] R. Vijay, "A Gentle Introduction to Natural Language Processing," Medium, Jul. 28, 2022. https://towardsdatascience.com/a-gentle-introduction-to-natural-language-processing-e716ed3c0863: :text=NLP%20is%20a%20branch%20of

[15] P. Huilgol, "Top 6 Open Source Pretrained Models for Text Classification you should use," Analytics Vidhya, Mar. 18, 2020. https://www.analyticsvidhya.com/blog/2020/03/6-pretrained-models-text-classification/: :text=Pretrained%20Model%20%231%3A%20XLNettext=

[16] K. Pham, "Text Classification with BERT," Medium, May 09, 2023. https://medium.com/@kha classification-with-bert-7afaacc5e49b: :text=How%20does%20the%20BERT%20model (accessed Oct. 07, 2023).

[17] vibhor nigam, "Natural Language Processing: From Basics, to using RNN and LSTM," Analytics Vidhya, Jan. 04, 2021. https://medium.com/analytics-vidhya/natural-language-processing-from-basics-to-using-rnn-and-lstm-ef6779e4ae66

[18] R. Winastwan, "Text Classification with BERT in PyTorch," Medium, Nov. 10, 2021. https://towardsdatascience.com/text-classification-with-bert-in-pytorch-887965e5820f

[19] M. Dutta, "Word2Vec For Word Embeddings -A Beginner's Guide," Analytics Vidhya, Aug. 02, 2022. https://www.analyticsvidhya.com/blog/2021/07/word2vec-for-word-embeddings-a-beginners-guide/

[20] A. R. Rout, "Advantages and Disadvantages of Logistic Regression," GeeksforGeeks, Aug. 25, 2020. https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/

[21] D. Han, "How to tackle dataset class imbalance for NLP," Medium, Jul. 01, 2023. https://medium.com/@dinghan1995/how-to-tackle-dataset-class-imbalance-for-nlp-4453af6f6b8

[22] https://www.facebook.com/jason.brownlee.39, "8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset," Machine Learning Mastery, Jun. 07, 2016. https://machinele to-combat-imbalanced-classes-in-your-machine-learning-dataset/

[23] T. Boyle, "Methods for Dealing with Imbalanced Data," Medium, Feb. 03, 2019. https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18

[24] A. Vidhya, "Imbalanced Classification | Handling Imbalanced Data using Python," Analytics Vidhya, Jul. 23, 2020. https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/

[25] L. Guadagnolo, "Imbalanced Data: an extensive guide on how to deal with imbalanced classification problems," Eni digiTALKS, May 03, 2022. https://medium.com/eni-digitalks/imbalanced-data-an-extensive-guide-on-how-to-deal-with-imbalanced-classification-problems-6c8df0bc2cab

[26] Anishnama, "Understanding Gated Recurrent Unit (GRU) in Deep Learning," Medium, May 04, 2023. https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2: :text=GRU%20stands%20for%20Gated%20Recurrent

[27] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," nlp.stanford.edu. https://nlp.stanford.edu/projects/glove/: :text=GloVe%20is%20an%20u

[28] "Gated Recurrent Unit Networks," GeeksforGeeks, Jul. 09, 2019. https://www.geeksforgeek recurrent-unit-networks/

[29] A. Mittal, "Understanding RNN and LSTM," Medium, Oct. 12, 2019. https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[30] A. Chugh, "Deep Learning | Introduction to Long Short Term Memory," GeeksforGeeks, Jan. 16, 2019. https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

[31] Vatsal, "Word2Vec Explained," Medium, Feb. 01, 2022. https://towardsdatascience.com/wo explained-49c52b4ccb71

[32] V. Reddy, “Deep Learning Vs Machine Learning with Text Classification,” Medium, Mar. 21, 2023. https://vasista.medium.com/deep-learning-vs-machine-learning-with-text-classification-162ea20a7924

[33] A. Biswal, “Power of Recurrent Neural Networks (RNN): Revolutionizing AI,” Simplilearn.com, Aug. 23, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn: :tex