Editorial Classification - Bloomberg

Aastha Joshi
aj2839@columbia.edu
Columbia University

Ameya Karnad
ak4251@columbia.edu
Columbia University

Nirali Shah
nss2173@columbia.edu
Columbia University

Sarang Gupta
sg3637@columbia.edu
Columbia University

Ujjwal Peshin
up2138@columbia.edu
Columbia University

Smaranda Muresan
sm761@columbia.edu
Columbia University

Daniel Preotiuc-Pietro
dpreotiucpie@bloomberg.net
Bloomberg LP

Kai-Zhan Lee
klee989@bloomberg.net
Bloomberg LP

Abstract

With the influx of myriad news in the Bloomberg terminal, distinguishing editorial news articles from regular articles is critical to aid its users in tailoring their news experience and further analyzing the impact of news on global financial markets. In this paper, we propose various *Machine Learning* and *Deep Learning* models to develop an editorial classifier that generalizes well across various news sources. The training set is made up of articles published by news sources from the US. We compare the performance of these models using the *Macro-Average F1-score* and *Matthews Correlation Coefficient (MCC)* as evaluation metrics to account for the presence of class imbalance in our data. Further, we gauged our models by comparing their performance on a Zero-Shot dataset which comprised of 1805 news articles published by *Metro Winnipeg*, a Canadian news source.

1 Introduction

Hundreds of thousands of news stories are published daily across the world. Bloomberg Terminal connects market participants to groundbreaking data through analytics and information-delivery service. This service is used by portfolio managers, investors, analysts, etc. to make investment decisions. The news stories are aggregated from multiple news sources around the globe. A significant portion of these news stories is subjective editorial and opinion content written by staffers or individual contributors with the goal of providing a personal viewpoint, insight or persuading the readers on a certain issue. Editorials shape the public discussion and offer commentary about the important events and issues of the time. They are written by regular columnists (editorials) and by external collaborators (op-eds) in an effort for most newspapers to diversify their opinions and offer a wider, more-specialized discussion on issues. Regular reports are filed by ground reporters and focus on precise factual details describing real-life incidents.

In contrast with factual news stories, editorials, op-eds, and commentaries are written with the goal of providing the reader with a subjective opinion or commentary, usually backed up by facts. Identifying these stories is useful for faceting and improving search and discoverability. It helps news consumers who are focused on a specific category to access articles pertaining to their interests.

Automatically studying the content of editorials and op-eds has applications in both computer science and social science. First, with the increase in online consumption of news through various channels such as social media. One form of online misinformation can be proliferated by presenting an opinion article as a factual article, lending the credibility of the source to the content. Another application is monitoring of the news or social media stream for potentially breaking news, and opinion articles should be discarded. It is thus paramount to be able to quickly and automatically detect stories from news sources that contain editorial and opinion content. For social science applications, changes in topics or emotional tone with time offer a new lens/perspective in content. Classification of news articles into various categories like regular, editorial, guest, op-ed, etc based on their content before they make

their way into the Bloomberg terminal is quite relevant. The information that is streamlined from the Bloomberg Terminal is used to make investment decisions. Having prior information about the category of an article is an important decision marker for the news consumer. This information can also be used by news publication houses to analyze the information being focused on currently and also to develop secondary product features like news recommendation for their product consumers. This information can also be used by researchers who aim to analyze patterns and trends in editorial content. The writing style and vocabulary used by authors of specific categories can provide useful insights to researchers too.

Several publications provide prospective op-ed with style guides on how to craft their stories. Hence, both due to the subjective and persuasive goal of the editorials and to their different styles, it is feasible to be able to use **N**atural **L**anguage **P**rocessing **(NLP)** and Machine Learning to automatically identify editorials from factual news stories. NLP offers powerful techniques for automatically classifying documents. These techniques are predicated on the hypothesis that documents in different categories distinguish themselves by features of the natural language contained in each document (Ramdass & Seshasai, 2009). Most salient features for document classification may include word structure, word frequency, and natural language structure in each document. The document structure plays an important role in ascertaining the category to which an article belongs to. We aim to use **NLP** based techniques to process the data from news articles and develop machine learning and deep learning models to classify it.

This paper presents the first NLP study of opinion content in newspapers. We have proposed models that are trained on historical archives of news articles. Most of these news articles were published within the span of 2018 - 2019. We use a corpus of stories identified as editorials through newspaper sections. We first build predictive models of editorials using a wide range of linguistics features computed over different sections of the editorials. We conduct an extensive linguistic analysis uncovering the content and style specific of editorials consistent across a wide variety of news sources. Our primary aim is to build a classifier of editorial and regular articles published across news sources. We use statistical techniques to model qualitative trends in language data. We show that using the content of the article, an F-1 macro score of 0.89 can be achieved using the **XLNet** model on the test data. Our results show that there are marked differences in textual features like word importance, character importance, and document context. Our findings reveal (Section 7) that quantifying these semantic metrics helps us identify the various categories to which news articles belong based on their content. We present our future work in Section 9.

2 Related Work

Recent studies have focused on analyzing the semantic types of claims and premises in online forums(Hidey, Musi, Hwang, Muresan, & McKeown, 2017). Many techniques focus on the identification of structural and lexical features that happen to be associated with persuasive arguments. Analysis of document structure and semantic features is helpful in extracting useful information about the text. Habernal and Gurevych (2016) use lexical features like verbs, tenses, sentiment scores, etc. and experiment with SVM and bidirectional LSTM to predict arguments scored by annotators as convincing. These methods use contextual references to model the data. They try to predict the convincingness of web arguments in an article pair. There has been research to model the features that explore the characteristics of communication that make someone an opinion leader.

Machine learning-based approaches are used for the automatic identification of discourse participants who are likely to be influencers in online communication (Biran, Rosenthal, Andreas, McKeown, & Rambow, 2012) Techniques to measure the variation in the contextuality of language (Heylighen & Dewaele, 2002) have also been studied. Nouns, adjectives, articles, and prepositions are more frequent in low-context or formal types of expression; pronouns, adverbs, verbs, and interjections are more frequent in high-context styles. This formed the basis for some of the feature engineering performed for our machine learning models. There has been work done to automatically detect evidence from unstructured text that supported a claim using machine learning-based approaches (Rinott et al., 2015). These techniques use context to detect evidence that can be used to support a claim in the discussion.

Modeling linguistic patterns has revealed that power differentials between participants are subtly revealed by how much one individual immediately echoes the linguistic style of the person they are responding to (Danescu-Niculescu-Mizil, Lee, Pang, & Kleinberg, 2012) This is relevant for our task as the authors for regular, editorial and other categories might have common subtle language markers across the multiple news sources from which the data has been collected.

3 Data

In this section, we talk about the Data and the news text and feature collection. We also mention the different programming packages and tools that were used for this task

3.1 Dataset

The data was obtained from Bloomberg. The initial data consisted of URL, News Source and Editorial labels. The editorial labels consisted of different categories such as Regular (regular news), Editorial, Oped, Guest (guest editorial), Roundup (multiple editors), and Others. The data obtained consisted of articles dated from 2008, but most of the articles were as recent as 2018 and 2019. The data consisted of news articles from 95 English news websites.

3.2 News text and features collection

As the initial dataset did not consist of the news content, article content had to be retrieved by extracting features from the HTML file of the URL present in the initial dataset. For this purpose, we decided to select a sample of 10 news sources on which initial news article text and feature collection would be performed. As all of the news sources had a high percentage of regular articles (It was noted that "Regular" labeled news items formed about 99.1% of all labels), it was decided that the data would be collected for the top ten news sources with a high count of minority labels. The news sources were,

- New York Times
- Washington Post
- Washington Observer Report
- Digital Journal
- Enid News
- Californian
- Press Democrat
- NW Florida Daily
- Gazette-Mail
- NJ Spotlight

The resultant data set had 35394 Articles. The articles are dated from 2013 onwards, but most of them are from the year 2018 and 2019.

4 Exploratory Data Analysis

We will explore different features of news articles in our dataset.

4.1 Categories of Articles

As mentioned before, the news articles in our dataset were had 6 categories: **Regular, Opinion, Oped, Editorial, Guest and Other**. To understand the distribution of news articles better, we visualized the *number of news articles* published in these categories across the 10 news sources we have worked on. The result is presented in *Figure 1*.

As is evident, most of the articles belong to the **Regular** category while few articles belong to the rest of the categories. This distribution prompted us to combine the 5 minor categories into a single category called **Non-Regular**. Another reason why the 5 minor categories were merged is because of the content of these article categories as described in section 1. All these categories had articles that

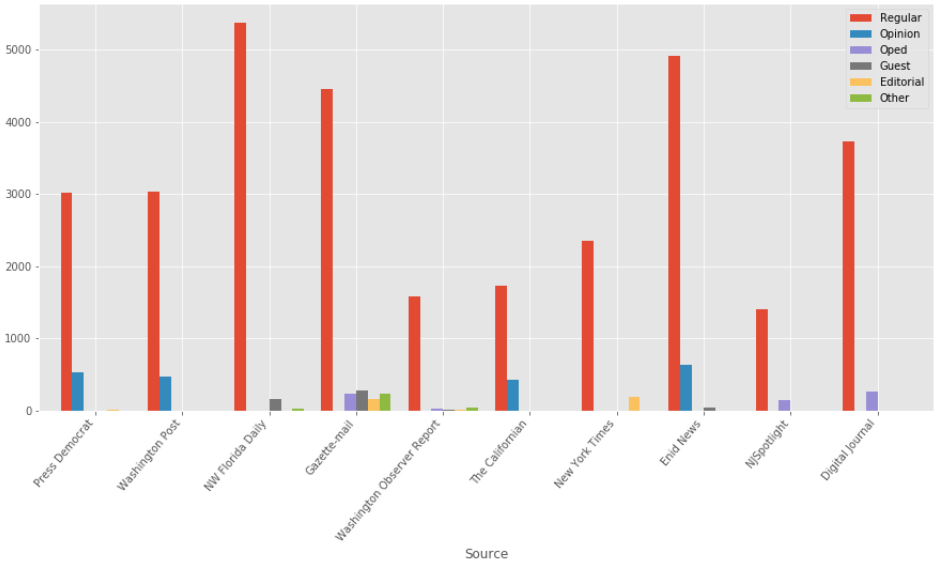

Figure 1: *Distribution of articles per category*

were opinionated. Regular category articles are published by ground reporters who reported on stories that are factual. The other categories, **Opinion, Oped, Editorial, Guest** and **Other** are all opinions of various people about different events that are based on individual perspectives. Consequently, we will be merging these five labels into the **Non-Regular** Class. Moving forward, we would refer to these 5 categories as part of the **Non-Regular** class unless otherwise specified. So ultimately, all the news articles would either be in the **Regular** class or **Non-regular** class

We visualized the newly categorized data across the news sources to explain the distribution of articles across the newly developed categories within each news source. The bar plot is presented in *Figure 2*.

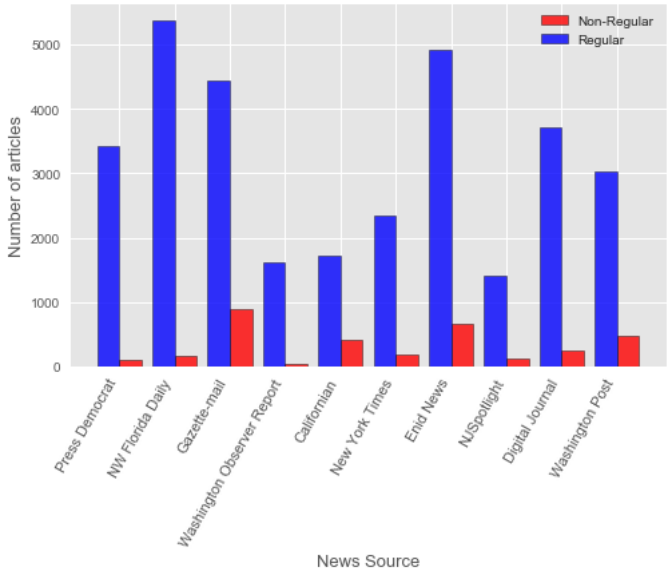

Figure 2: *Distribution of Articles - Regular and Non-Regular*

As anticipated, there exists a heavy imbalance in the data with **Regular** being the major category. To build a model that can accurately classify news articles, it is important to have a balanced dataset to ensure that the model does not predict the majority class by default. This visualization was a key factor for us to confirm that we needed to use techniques like *downsampling* or *upsampling* before developing a model.

4.2 Count of Words

To clean the dataset based on the length of news articles, we developed a visualization to depict the distribution of article length across news sources, the result of which is presented in *Figure 3*.

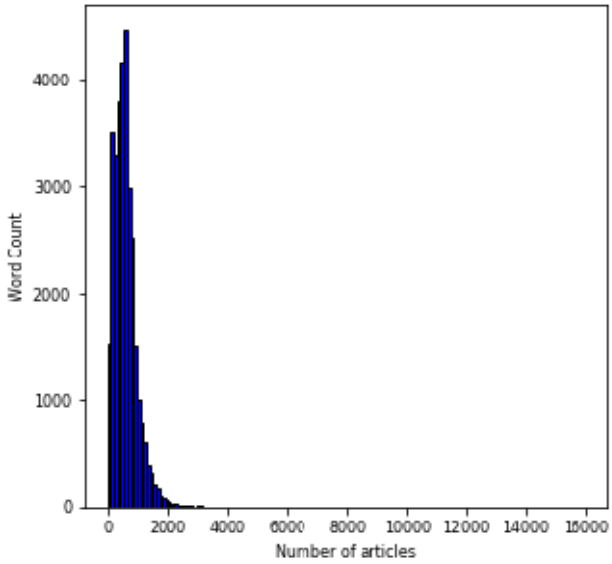

Figure 3: *Distribution of Article Length*

As seen in *Figure 3*, the *95% confidence interval range for article length* is [100,2000]. Few articles were more than 2000 words long and very few of them had a word count greater than 3000. This derived insight prompted the elimination of articles with word count outside the 95% confidence interval.

As a part of further inquisition, we analyzed the *word count distribution* of articles for each news source present in our dataset to figure out if the distribution shown in Figure 3 was heavily impacted by some news sources.

We discovered that *The New York Times* generally publishes articles that are up to 4000 words long while most of the articles published by the other news sources are comparatively shorter with a word count of up to 2000. This analysis justified that we could clean the dataset based on the word count of articles without losing too much information.

4.3 Feature analysis

Figure 4 shows the word cloud generated for news articles belonging to the **Regular** and **Non-regular** categories. The entire corpus of training data was used as a document corpus to generate this visualization. In order to extract the words and give them weighting based on their counts, scikit-learn's CountVectorizer was used. The top 50 most common non-stopwords were extracted for each of the 2 categories of news articles.

It can be seen that news articles belonging to the regular category have a common occurrence of words like time, school, state, etc . Whereas for news articles of non-regular category words like trump, president, state, etc. are more commonly used. This analysis corroborates with the fact that non-regular articles tend to focus more on articles related to politics and current affairs where the author depicts his/her opinion for a given situation. Authors of regular articles are more likely to focus on a variety of topics mostly focusing on the current happening on the ground.

4.4 Part-of-speech Tagging (POS)

Part-of-speech (POS) tagging is the process of assigning a word in a corpus as corresponding to a part of speech such as 'Nouns', 'Pronoun', 'Adverb', 'Adjective' etc. We explore the presence of different POS tags in the two classes.

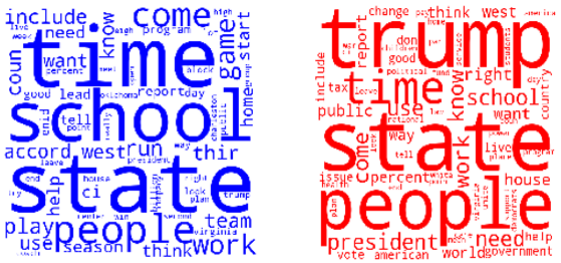

Figure 4: *Word Cloud - Regular and Non-Regular*

To compare the two classes, we first calculate an *Instance per article length* metric for each POS tag which evaluates the percentage of tokens corresponding to those POS tags in the article. We then look at the distribution of *Instance per article length* in the two classes. Our general hypothesis is that the distribution of various POS tags used in **Regular** and **Non-Regular** articles would vary. We use the *spaCy* library in *Python* for POS retrieval. The result of our exploratory analysis of POS is presented in *Figure 5*. As expected, most of the graphs follow a normal distribution, but there are some striking differences in the distribution of parts of speech in the two classes. Parts of speech like 'Adjective' and 'Adverbs' seem to be used more frequently used in **Non-Regular** articles than in **Regular** articles, perhaps due to the fact that **Non-regular** articles are more opinionated than **Regular** articles and provide a more "descriptive analysis" of the 'Nouns' (Person/Topic of interest) and 'Verbs' (Action of interest), than factual news. On the other hand, there are more mentions of 'Numbers' and 'Proper-Nouns' in **Regular** articles than in **Non-Regular** Articles. This makes intuitive sense as 'Numbers' and 'Quantities' convey hard facts.

*Note: *The bins in the graphs are left exclusive (i.e the first bin does not include the value 0), So all articles with instance per article length 0 for a particular part of speech will be discarded from the graph. Keys for each of these POS label are present in the Appendix section of the report*

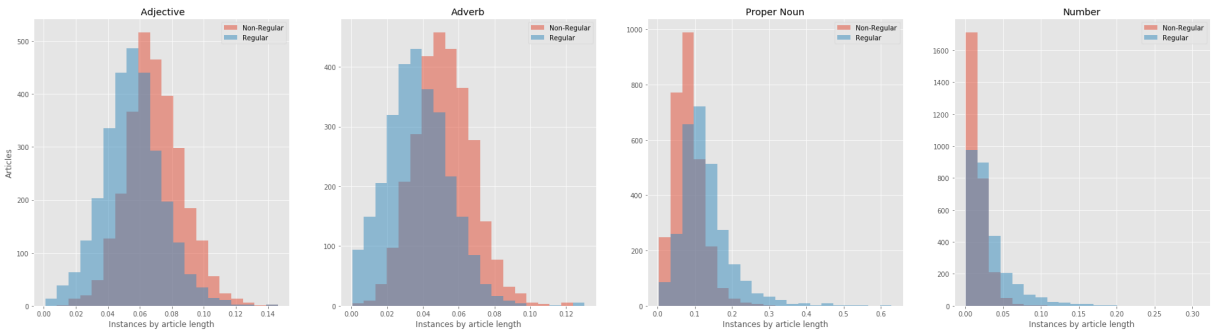

Figure 5: *Distribution of Parts-of-Speech Tags*

4.5 Named-entity Recognition (NER)

Named-entity Recognition (NER) aims to recognize mentions of rigid designators from text belonging to predefined semantic types such as 'Person', 'location', 'organization' etc (Nadeau & Sekine, 2007). Similar to the part-of-speech analysis, we use the NER implementation provided in the *spaCy* package. We calculate the *Instance per article length* for each *named entity* in the two classes.

As can be seen from Figure 2, graphs for Named Entity Recognition were heavily right-skewed for most of the entity types. **Non-Regular** Articles have higher mentions of 'Law' and 'Nationalities' entities as compared to **Regular** articles in the article text. On the other hand, 'Locations', 'Product', 'Quantity', 'Time' and 'Date' have more a prominent presence in **Regular** articles.

*Note :*The Keys for each of these Named Entity Recognition labels are present in the Appendix section of the report*

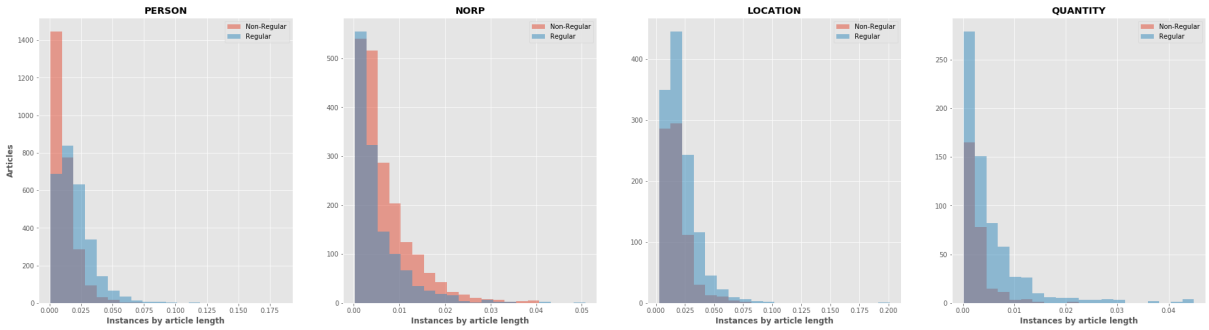

Figure 6: *Distribution of Named-entity Recognition Tags*

4.6 Correlation Analysis

Several other features were extracted from both raw articles, processed text (explained in Section 5), and headlines. General-purpose text features included *average word length* and *sentiment* (using VADER (Hutto & Gilbert, 2014), AFINN (Nielsen, 2011a), and TextBlob) of the article (both retrieved from the processed text). Other features retrieved from the raw articles include *heading length*, *article length*, *number of punctuation* (question marks and exclamations), *subjectivity* (TextBlob).

We conducted a covariance analysis to examine how different features of the articles relate to each other and the target variable. Some significant correlations are visible in the matrix (Insert Figure). For instance, the POS tag for 'Proper Nouns' is significantly correlated with 'Person' and 'Organization' NER tag. This makes sense as person and org names are more likely to be proper nouns than common nouns. There are some significant correlations between article features and the target variable as seen in figure 7(a). Opinion articles, in general, have a higher number of questions (normalized for article-length) as compared to regular articles (Figure 7(b)). Similarly, regular articles are longer in length than opinion articles.

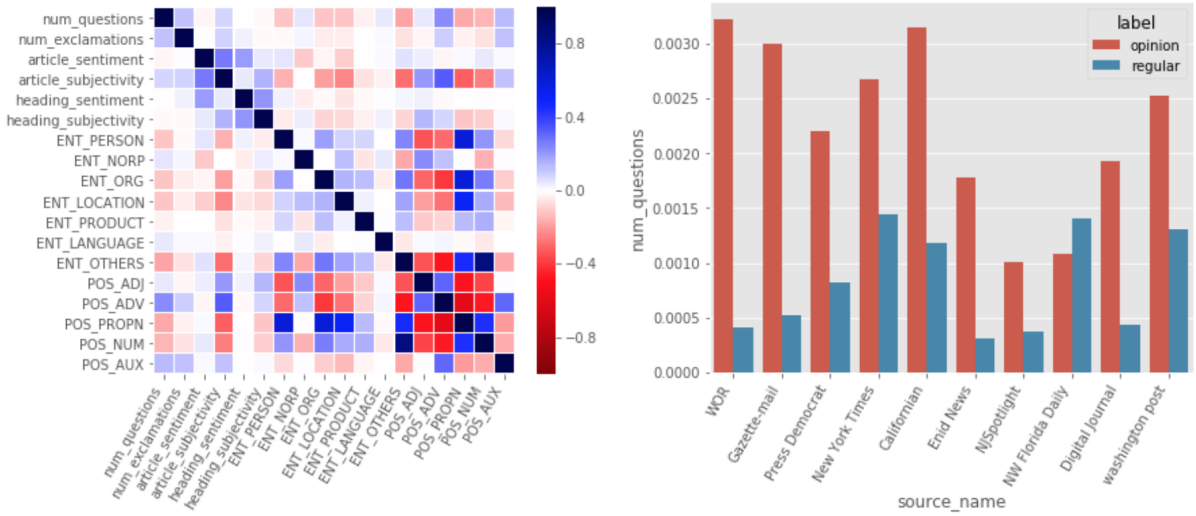

Figure 7: *a) Correlation between features b) Distribution of number of questions by class and source*

5 Data Preparation

Data preparation is a crucial part of the project. This step works on the data to create features and processed-text that can be used by Machine learning and Deep learning models. The data preparation module used a two-pronged approach to get the data ready:

5.1 Train-test split and class imbalance handling

The train-test split are carried out in a temporal fashion. The split it similar to the *TimeSeriesSplit* in scikit-learn where the testing data is in the future of the training data. This was thought of a simulation of a production setting, where a model trained on a few days old data can perform well and does not need to be updated day and again. If we take a 90-10 train-test split, we would calculate a date such that 90% of the data before it is train and 10% after it is test.

After finding that split, another issue is the balance of the dataset. The original dataset has a balance of 89-11 (**Regular - Non-Regular**). Hence, as mentioned in Section 4, we need to undersample the **Regular** class. We undersampled the data in such a fashion such that, for each instance of **Non-Regular** class in our dataset, we would have another instance of **Regular** class from the same news source and from the same date. If there were no articles of **Regular** class available on the same day, we applied a lookback-lookforward paradigm, where you can pick another article within a lookback and lookward period. As lookforward could lead to potential data leak, we only used lookback of 7 days in our data.

Hence, the training data had 6386 articles (3193 **Regular** and 3193 **Non-regular**) and the test set had 3429 articles (3076 **Regular** and 353 **Non-regular**)

5.1.1 Text Processing and Feature Creation

Additional preprocessed-text was generated, which would aid the generalizability of the model. We took a standard approach to preparing the preprocessed text, which included steps such as, expanding contractions to their full forms using a dictionary of normal contractions, converting to lower-case, removing punctuation, processing HTML, removing URLs from articles, removing stutterings, removing very short words, removing stop words, and Wordnet lemmatization (Miller, 1995).

The features were created using both the original text and the preprocessed text. General-purpose text features included Article and Heading lengths, Sentiments (VADER (Hutto & Gilbert, 2014), AFINN (Nielsen, 2011b), TextBlob), Punctuations (exclamation and question marks), Subjectivity - (TextBlob), Counts of Named Entities and Parts-of-Speech, Bag-of-words on word-level(unigram, bigram, trigram) character-level(unigram and bigram), TF-IDF on word-level and character-level

6 Methods

6.1 Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable (Regular/ Non-Regular in our project). We developed a *Logistic Regression* model using a count vectorizer as a baseline. Count Vectorizer provides a simple way to both tokenize a collection of news articles and build a vocabulary of known words, but also to encode new articles using that vocabulary. This baseline model was trained on the first 64, 128, 256 ad 512 words.

Further, we developed a *Logistic Regression model* using *Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer* on *unigrams and bigrams* (ngram_range = (1,2)). TF-IDF is a numerical statistic that is intended to reflect how important a word is to an article in a collection of articles. The TF-IDF value increases proportionally to the number of times a word appears in the article and is offset by the number of articles in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. The only parameter tuned for this model was *C*.

Based on the results shown in 5 and 6, we also trained a Logistic Regression model on the count of different Named Entity Recognition and Parts-of-Speech features detected in the news articles. The parameter *C* was tuned to obtain better results.

6.2 Decision Tree

One of the standard classifiers we tried was the *Decision Tree Classifier*. A decision tree is a flowchart-like tree structure where an internal node represents a feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. It artificially performs complex decision making. One of the reasons why Decision Tree is so powerful is because it mimics human thinking, thereby making it easily interpretable.

We trained a Decision Tree Classifier on TF-IDF vectors generated on unigrams and bigrams from the article text. This model was trained for different sequence lengths viz, 64, 128, 256, 512 and the parameter tuned in each of these individual models was *max_depth*.

6.3 Random Forest Classifier

Random Forest is an *ensemble learning* method for classification which operates by constructing a multitude of decision trees on various sub-samples of the dataset and aggregating votes from these trees to decide the final class of the test object. This aggregation limits overfitting, as well as error due to bias and thus, is more robust than a single decision tree.

As in case of all the models above, we ran this model using TF-IDF Vectorizer and n_gram range = (1,2) on the first 64, 128, 256 and 512 words of the article. To obtain better results, we tuned the parameters *max_depth* and *n_estimators*.

6.4 LightGBM

Along with other standard classifiers, we also tried LightGBM (Ke et al., 2017). It is frequently used in Kaggle competitions, trains fast and gives comparable performance to other ensemble models. It is a model that is primarily based on decision trees. It splits the trees leaf-wise instead of tree-depth wise or level-wise. Hence, it leads to faster training time and more reduction in loss (Al Daoud, 2019).

LightGBM uses Gradient-based One-Side Sampling to find which value to split on, which is faster than a Histogram-based approach for finding splits. So, it keeps all instances with higher gradients, and also randomly samples instances which have a lower gradient. The assumption here is that those samples with lower gradient have a smaller training error, and hence the model is already tuned to those samples. The parameters tuned for LightGBM were max_depth, num_leaves, n_estimators, and learning_rate.

6.5 LSTM, BiLSTM and BiLSTM-Attn

The first neural model is a **L**ong-**S**hort **T**erm **M**emory **(LSTM)** network (Hochreiter & Schmidhuber, 1997). Tokens ti in a given article T = t1, ..., tn are computed for different article lengths viz. 64, 128, 256 and 512. These tokens are mapped to pre-trained **G**love (Pennington, Socher, & Manning, 2014) embeddings and passed through an LSTM layer. This is then passed through a global MaxPooling layer followed by a Dropout layer. Dropout is used for model regularisation and to prevent overfitting. Finally, a Dense layer with softmax activation function is used to compute a probabilistic output score.

Next, a Bidirectional LSTM (Schuster & Paliwal, 1997) layer was used to model temporal dependencies. BiLSTM connects 2 hidden layers of opposite directions to the same output. Outputs from both states are not connected but concatenated at each time step. This helps the model to learn additional context.

On further experimentation with BiLSTM, word-level attention((Yang et al., 2016) was incorporated. A single vector is computed as a sum of resulting contextualized vector representations ($\sum_i aihi$) at every timestep t. This is subsequently passed through the final prediction layer.

Next, a Multi-Input BiLSTM model was developed. The 2 inputs are article heading and text. Both input sources were trained separately and the resultant vectors were concatenated. Subsequently, the final vector was used to predict the class probabilities.

6.6 BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) is a deep language understanding model based on transformer networks (Vaswani et al., 2017) pre-trained on large corpora (Devlin, Chang,

Lee, & Toutanova, 2018) released by Google AI in 2018. The model makes use of multiple multi-head attention layers to learn bidirectional embeddings for input tokens. It is trained for masked language modeling, where a fraction (15%) of the input tokens in a given sequence are masked and the task is to predict the masked word given its context. BERT uses wordpieces which are passed through an embedding layer and get summed together with positional and segment embeddings. The former introduces positional information to the attention layers, while the latter contains information about the location of a segment.

Similar to the previous model, we run all 4 different BERT models (BERT base-cased, BERT base-uncased, BERT large-cased, BERT large-uncased) on sequences of lengths 64, 128, 256 and 512. Hugging Face's BERT implementation (Wolf et al., 2019) in PyTorch was used.

6.7 XLNet

XLNet (Yang et al., 2019) is a model that was released by CMU and Google Brain in conjunction in June 2019. It claimed to outperform BERT, which was the previous state-of-the-art in 20 language modeling tasks.

XLNet is a generalized Autoregressive model, that uses either forward or backward context in order to train the model. XLNet allowed Autoregressive models to learn from forward as well as backward context at the same time. XLNet defined a new training objective called Permutation Language Modelling, which was based on permutations. During training, sentence words are permuted and the model predicts a word given the shuffled context. Similar to BERT, XLNet was trained on base-uncased, base-cased, large-uncased, and large-cased, on sequences of lengths 64, 128, 256, and 512, where the only limitation was that the XLNet large models could not be trained on length 512 due to computation limitations.

7 Results

In this section, we evaluate the models based on certain metrics and discuss some salient findings from the models. We discuss the metrics that we have used to evaluate the models as well as some of the interesting results that we obtained from model runs.

7.1 Evaluation Metrics

A complete list of evaluation metrics that were used to establish a comparison between the models are listed in the Appendix. Below, we discuss two of the metrics that enable us to take class imbalance into consideration while evaluating the models.

7.1.1 Matthews Correlations Coefficient (MCC)

Introduced by Matthews (1975), the *Matthews Correlation Coefficient* takes all the four quantities of a 2x2 confusion matrix into account while calculating the metric (as compared to *precision* and *recall*). This lends the metric particularly useful when evaluating the performance of model trained on imbalanced classes. It can be calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$$

7.1.2 Macro-Average F-1 Score

Similar to MCC, the *Macro-Average F-1 score* is a useful metric when the class is highly imbalanced. *Precision* and *Recall* is computed independently for each class and averaged out before calculating the *F-1 score*. Both classes are given equal weights, as is the case in our classification task. The quantities can be calculated as follows:

$$F1(Macro) = 2 \times \frac{Precision(Macro) \times Recall(Macro)}{Precision(Macro) + Recall(Macro)}$$

$$Precision(Macro) = \frac{(\frac{TP}{TP+FP}) + (\frac{TN}{TN+FN})}{2}$$

$$Recall(Macro) = \frac{(\frac{TP}{TP+FN}) + (\frac{TN}{TN+FP})}{2}$$

7.2 Prediction Results

Figure 8 shows the performance of different models on the aforementioned metrics. Performance on the best hyperparameter setting for sequence lengths 64, 128, 256 and 512 are plotted for each model.

For the sake of reporting, the best models out of different types of models were selected. The best model among the LSTM models was the BiLSTM model. The BERT base-uncased model performed the best among all the BERT models. The BERT Large models provided promising results for small sequence lengths, but could not run for larger sequence lengths. Meanwhile, XLNET base-cased worked the best among XLNET models. Similar to BERT Large models, XLNET large models could not run for high sequence lengths.

The reported graphs show that most models perform better at higher sequence lengths. Linear models could run for higher sequences but we decided to restrict at 512 sequence length for the scope of this project as deep Learning models failed to run for higher sequence lengths due to lack of computation resources.

We found out that XLNet is the best performing model on both the Macro Average F-1 Score metric (Figure 8 (a)) and Matthew's Correlation Coefficient metric (Figure 8 (b)) followed by BERT. This is most likely due to the fact that XLNet and BERT are pre-trained on a large corpus of English text, which enables it to understand the semantics of the language. Moreover, they have complex architectures that enable them to learn the underlying structure of the data as compared to the linear models. All the other models, on the other hand, have been trained from scratch and do not provide the benefits associated with transfer learning.

Furthermore, Logistic Regression models fitted with count vectorizer (baseline) and Tf-IDF vectorizer performed better than the three tree-based models.

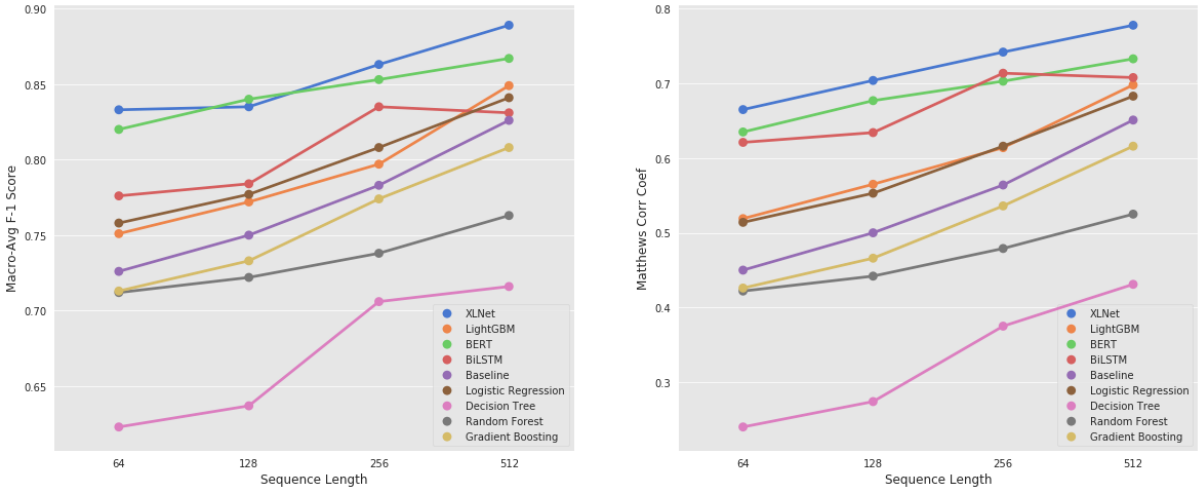

Figure 8: *(a) Performance of different models with their Macro Avg F1 Scores. (b) Performance of different models with their Matthews Corelation Coefficients.*

The comprehensive results can be found on the github link:
`https://github.com/ujjwal95/bloomberg_editorial_classifier/tree/master/etc/results`

7.3 Evaluation on External Data

The training and test data originated from the same 10 new sources. In order to evaluate the generalizability of our model, we ran the best configurations of each of the models on a zero-shot dataset. Articles from the Canadian Newspaper *The Metro Winnipeg Free Press* was chosen as the zero-shot dataset. This

News Source was specifically selected as it a non-US based news source and would allow us to test if the models are generalizable to texts with lexical and topical differences (such as people, locations) with our training data.

The number of articles in the zero-shot dataset were 1805 (1657 **Regular** and 148 **Non-regular**)

Figure 9: *Macro-Average F-1 Score*

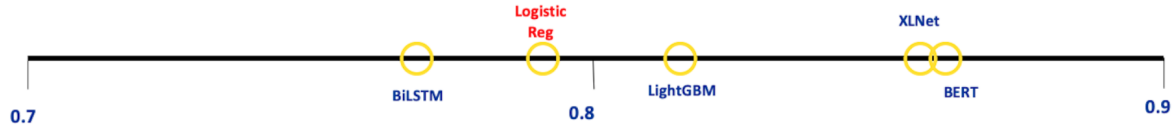

Figure 10: *Matthew's Correlation Coefficient*

As it can be seen from *Figure 9* and *Figure 10*, the MCC and Macro-Average F-1 scores are fairly representative of the results obtained on the original validation data. BERT performs the best on the two metrics followed by XLNet. Machine learning models such as Logistic Regression and LightGBM have similar performance to on the validation and the external set. These results imply that the models can be used in a general setting to classify English new articles regardless of their origin.

7.4 Other Results and Interpretations

7.4.1 Feature Importance - Linear Models

The feature importances generated by the linear model (Figure 11) helps us determine the most important features for this classification task. While the most important features varied across different models, the most persistent ones included 'ENT_OTHERS' (Other entities), POS_PROPN (Proper nouns), POS_ADV (Adverbs), NEU_VADER (Sentiment). Specific words like 'said' and other characters identified by the TF-IDF vectorizer also had high feature importance.

7.4.2 Attention in BiLSTM

Adding attention to our BiLSTM model did show a marginal macro F1 score improvement for word length 128 and 256. However, a slight decrease in the Macro Average F1 score was observed for the word length of 64 and 512. The visualization for the attention for an opinion article with article length as 256 and 64 hidden units of the BiLSTM model is as in *Figure 12*.

It can be seen from *Figure 12* that the model pays more attention to words like why, don't, lie, all, etc. which convey strong emotions of the author. Words like 'both', 'guys', 'democracy', 'die', etc. are given a lower score as they are more generic and don't give any strong opinions of the author.

8 Conclusion

Our aim was to build an editorial classifier that can categorize articles from various news sources across the globe as Regular or Non-Regular. To this extent, we evaluated the performance of our machine learning and deep learning models on a Zero-shot dataset which comprised of *1805* news articles published by *Metro Winnipeg*, a *Canadian* news source. This was done to ensure that our models do not categorize articles based on the most frequently occurring topics in articles published by local news sources. While XLNET performed the best on our test data set, closely followed by BERT as can be seen in Figure 8, BERT outperformed XLNET on the Zero-Shot dataset by a marginal amount. The Macro- Average

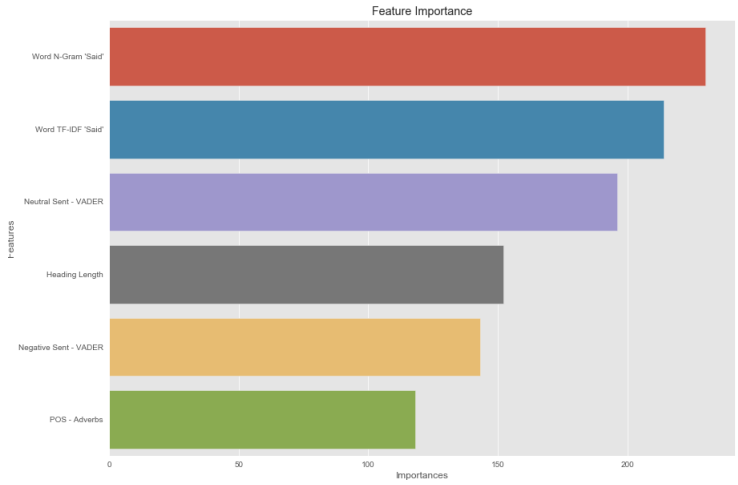

Figure 11: *Feature importance as per Logistic Regression Model*

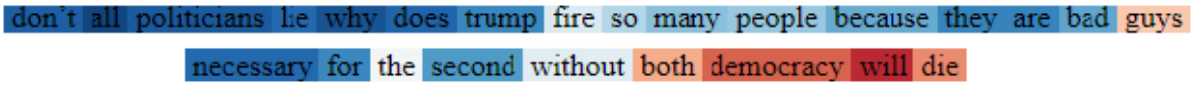

Figure 12: *BiLSTM attention based word score*

F1-score was 0.932 for the BERT model and 0.930 for the XLNET model. An interesting thing to note was that a simple Logistic Regression model with TF-IDF vectorizer and n_gram range = (1,2) had a macro-average F1-score of 89.6.

The main takeaway from this paper is that for better results, it is advisable to use BERT and XLNET. But in the case of lack of Hardware and computational resources, it is recommended to use Logistic Regression as the model's performance is quite comparable to the deep learning models and unlike deep learning models, it does not lose out on model explainability. Moreover, Logistic regressions are able to incorporate the entire sequence for prediction, which may imporve their performance. Moreover, Logistic Regression models can incorporate the entire sequence for prediction which increases the chance of better prediction as shown in Sections 7

9 Future Work

With the news article data source that is currently available, there is an opportunity to explore more in the domain of News media.

9.1 Multi-class Classification

The scope of our project was currently restricted to a binary classification of News articles into Regular and Non-Regular categories. But as future scope of this project, we would be looking to explore the Non-Regular class more and design a multi-class classification system to classify articles into **Opinion, Oped, Editorial, Guest and Other** classes. As all the Non-Regular class articles are 'opinionated' article, we would like to explore other features that would differentiate between these labels

9.2 Fine-grained editorial classification

Currently, our model looks to classify the entire news article. As a future scope of this project, we would like to extend this project to classify the articles on a paragraph and sentence levels. This may help a ground-reporter to ensure that the news article that he/she is publishing is not perceived to be 'opinionated'. This will also help differentiate "Facts" from "Opinions" in any article.

The challenge this task entails is the lack of data in the small parts of the news article. Hence we would have to develop efficient algorithms that can classify text with less amount of data

9.3 Topic Modeling

It would be useful to discover hidden semantic structures in article content using Topic Modeling. Developing a topic model can aid us in extracting abstract topics from a collection of articles. To perform topic modeling, we can use Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2002). LDA is based on the assumption that each document (news article) in a corpus is a combination of topics inferred across all the documents. The topic structure is unknown, and it observes documents and their content and uses it to infer the topic structure(Jacobi, Van Atteveldt, & Welbers, 2016). Using this technique, we can figure out if some news sources and authors tend to publish articles based on a certain set of topics. This can also help us to figure out the most common topics as well as the most frequently used words within the topic.

A time dimension extension of LDA can also be used to see how keywords in the same topic changed over time. Dynamic topic models (Blei & Lafferty, 2006) can be used to analyze the evolution of latent topics of a collection of documents over time. This can help us determine the most important talking points and keywords during a particular period of time.

9.4 News Sentiment Analysis

It would be interesting to perform sentiment analysis or opinion mining to determine the sentiment of the news article. Sentiment analysis would also be able to determine the emotional state of the writer. Currently, VADER (Valence Aware Dictionary and sEntiment Reasoner) (Hutto & Gilbert, 2014), which is a part of the Natural Language Toolkit (NLTK) is a rule-based sentiment analysis tool that can help detect if the text has a positive, negative or neutral sentiment. An enhancement of this model can be built by manually tagging different articles with various sentiments and emotions using Amazon Mechanical Turk. Supervised Machine learning and Deep learning models can then be used to help classify the data with a multi-label classification algorithm. This can also be used to model how different news sources frame their content for maximum readership impact.

9.5 Political Bias and Fake News Detection

The rise in the accessibility of the internet and the boom of social media has made content creation simple (Baly, Karadzhov, Alexandrov, Glass, & Nakov, 2018). This has made it easier for everybody to share and spread information online. While this has helped crowdsource journalism and has bought important local news to the foreground, it has left the public unprotected against fake news and misinformation campaigns. Fake news has now meddled with a US Presidential election (Allcott & Gentzkow, 2017), lead to mob lynchings in India (Arun, 2019; Bali & Desai, 2019), affected stock market (Gingerich, 2019) and cause diplomatic incidents (Jazeera, 2018; Filho, 2018).

Hence there is an need to build supervised machine learning models to identify fake news. Amazon Mechanical Turk can be used to tag real and fake news from given dataset. Later, NLP based Machine learning and Deep Learning discussed in section 1 can be used to trained and build models to detect fake news.

Similarly, political bias and partisanship in media has also increased in the past few years. Editors of different News media houses are looking to influence people with their points. Hence political bias in editorial articles can also be detected using NLP, supervised machine learning and deep learning techniques. We could also look at how people from different sides speak on similar issues differently to convince their readers of their point. This editorial article analysis can be done using an approach similar to the one used by Habernal and Gurevych (2016).

Acknowledgements

We would like to express our special thanks of gratitude to our mentors from Bloomberg LP, Daniel Preotiuc-Pietro and Kai-Zhan Lee for the opportunity to work with them. We are thankful to them for

their guidance throughout the project, and for patiently answering our queries no matter how small they were. We would also like to convey our gratitude to our faculty supervisor, Prof. Smaranda Muresan, whose expert advice on Natural Language Processing helped us throughout the project.

Lastly, we would like to thank the Data Science Institute, Capstone Project DSI facilitator Prof. Eleni Drinea, and Capstone Instructors Professors Sining Chen and Adam Kelleher for providing us an opportunity to work in an exciting project with these wonderful mentors.

References

Al Daoud, E. (2019). Comparison between xgboost, lightgbm and catboost using a home credit dataset. *International Journal of Computer and Information Engineering*, *13*(1), 6–10.

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of economic perspectives*, *31*(2), 211–36.

Arun, C. (2019). On whatsapp, rumours, and lynchings. *Economic & Political Weekly*, *54*(6), 30–35.

Bali, A., & Desai, P. (2019). Fake news and social media: Indian perspective. *Media Watch*, *10*(3), 737–750.

Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., & Nakov, P. (2018). Predicting factuality of reporting and bias of news media sources. *arXiv preprint arXiv:1810.01765*.

Biran, O., Rosenthal, S., Andreas, J., McKeown, K., & Rambow, O. (2012). Detecting influencers in written online conversations. In *Proceedings of the second workshop on language in social media* (pp. 37–45).

Blei, D. M., & Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on machine learning* (pp. 113–120).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2002). Latent dirichlet allocation. In *Advances in neural information processing systems* (pp. 601–608).

Danescu-Niculescu-Mizil, C., Lee, L., Pang, B., & Kleinberg, J. (2012). Echoes of power: Language effects and power differences in social interaction. In *Proceedings of the 21st international conference on world wide web* (pp. 699–708).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Filho, J. (2018, Jan). *Brazilian right-wing fell hard for a fake news story about venezuela, provoking diplomatic incident.* The Intercept. Retrieved from `https://theintercept.com/2018/01/15/fake-news-brazil-venezuela/`

Gingerich, J. (2019, Nov). *The cost of fake news: $78 billion.* Retrieved from `https://www.odwyerpr.com/story/public/13448/2019-11-26/cost-fake-news-78-billion.html`

Habernal, I., & Gurevych, I. (2016). Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1589–1599).

Heylighen, F., & Dewaele, J.-M. (2002). Variation in the contextuality of language: An empirical measure. *Foundations of Science*, *7*(3), 293–340.

Hidey, C., Musi, E., Hwang, A., Muresan, S., & McKeown, K. (2017). Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th workshop on argument mining* (pp. 11–21).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hutto, C. J., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international aaai conference on weblogs and social media.*

Jacobi, C., Van Atteveldt, W., & Welbers, K. (2016). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, *4*(1), 89–106.

Jazeera, A. (2018, Feb). *Fake news and the gulf crisis.* Author. Retrieved from https://www.aljazeera.com/programmes/insidestory/2018/02/fake-news-gulf-crisis-180216212644373.html

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (pp. 3146–3154).

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, *405*(2), 442–451.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Nielsen, F. Å. (2011a). Afinn. *Richard Petersens Plads, Building*, *321*.

Nielsen, F. Å. (2011b). A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Ramdass, D., & Seshasai, S. (2009). Document classification for newspaper articles. *Document classification for newspaper articles*.

Rinott, R., Dankin, L., Perez, C. A., Khapra, M. M., Aharoni, E., & Slonim, N. (2015). Show me your evidence-an automatic method for context dependent evidence detection. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 440–450).

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... others (2019). Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies* (pp. 1480–1489).

Appendix

Part of Speech Keys

- 'ADJ' : Adjective - word that modifies/describes a noun or noun phrase
- 'ADP' : Adposition - cover term for prepositions and postpositions
- 'ADV' : Adverb - word that modifies/describes a verb
- 'AUX' : Auxiliary verb - verb that adds functional or grammatical meaning to the sentence
- 'CONJ' : Coordinating conjunction - a conjunction placed between words, phrases, clauses, or sentences
- 'DET' : Determiner - word that serves to express the reference of a noun
- 'INTJ' : Interjection - Word or expression that expresses a spontaneous feeling or reaction
- 'NOUN' : Noun - word that functions as the name of some specific thing or set of things
- 'NUM' : Numeral - Numbers and figures
- 'PART' : Particle - word with traditional meaning
- 'PRON' : Pronoun - word that substitutes for a noun or noun phrase
- 'PROPN' : Proper noun - noun that identifies a single entity
- 'PUNCT' : Punctuation - punctuation marks such as comma, period, parentheses, etc.
- 'SCONJ' : Subordinating conjunction - a conjunction that introduces a subordinate clause
- 'SYM': symbol - Special Symbols
- 'VERB' : verb - word used to describe an action, state, or occurrence

Named Entity Recognition Keys

- 'PERSON': People, including fictional.
- 'NORP': Nationalities or religious or political groups.
- 'ORG': Companies, agencies, institutions, etc.
- 'LOCATION': mountain ranges, bodies of water, counter, cities, states, buildings, airports, highways, bridges, etc.
- 'PRODUCT': Objects, vehicles, foods, etc. (Not services.)
- 'EVENT': Named hurricanes, battles, wars, sports events, etc.
- 'WORK_OF_ART': Titles of books, songs, etc.
- 'LAW': Named documents made into laws.
- 'LANGUAGE': Any named language.
- 'DATE': Absolute or relative dates or periods.
- 'TIME': Times smaller than a day.
- 'PERCENT': Percentage, including "
- 'MONEY': Monetary values, including unit.
- 'QUANTITY': Measurements, as of weight or distance.
- 'ORDINAL': "first", "second", etc. 'CARDINAL': Numerals that do not fall under another type.

List of Evaluation Metrics

- Precision
- Recall
- TNR
- Accuracy
- Balanced Accuracy
- F1 Score
- Matthews correlation coefficient

- Macro average precision
- Macro average recall
- Macro average F1 Score
- Weighted average precision
- Weighted average recall
- Weighted average F1 Score