SENTIMENT ANALYSIS OF NEWS ARTICLES USING BIDIRECTIONAL RECURRENT NEURAL NETWORKS

ALEXANDER TAN KA JIN

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Your Degree

Faculty Of Computing
Universiti Teknologi Malaysia

21/12/2024

Exploratory Data Analysis

Data set

All-The-News 2.0 is a collection of 2.6 million news articles from US based news sites such as Reuters, The New York Times and CNN.(Thompson, n.d.) It also includes tabloids and magazines such as TMZ and Vox within it's data. The news articles are scraped from news sites using python webcrawlers and is free to be downloaded and used for analysis outside of use for training generative AI. (Thompson, n.d.) The data is compiled from 26 news sites with the majority of data originating from Reuters. A huge amount of articles are on the shorter side being less than 22479 characters long.

The initial data contained 10 columns detailing the data, year, month, day, author,

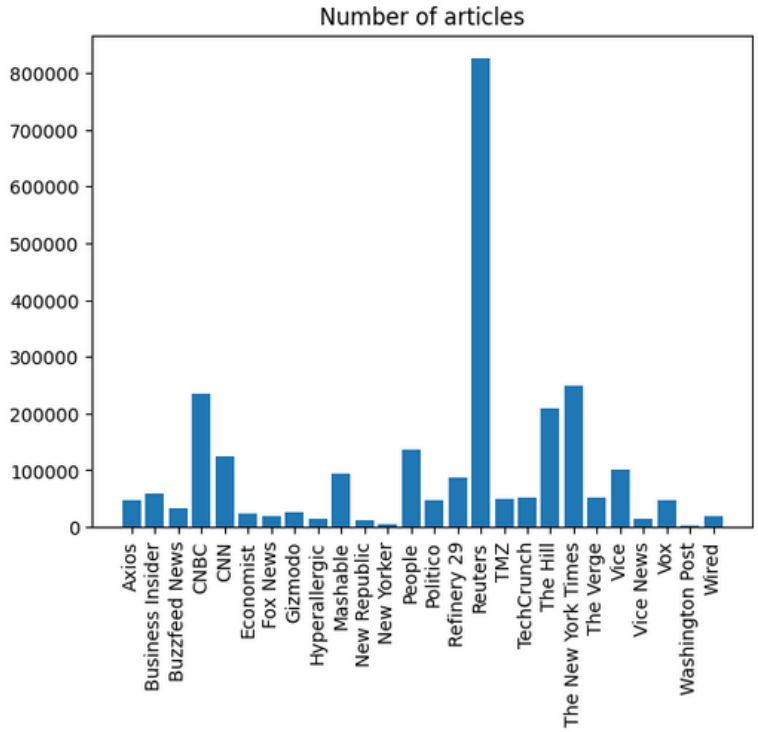

Figure 1 Data article count sorted by publication.

title, article text, url, publication section and publication sites. It includes news detailing current events and politics like Reuters and also tabloids/pop culture and tech magazines like TMZ or TechCrunch which the latter is not relevant to the thesis.

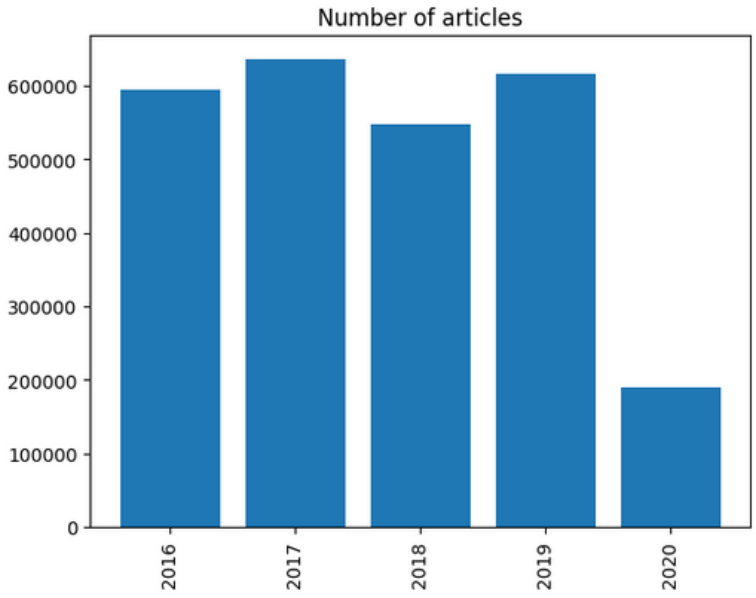

Figure 2 Data article count sorted by year.

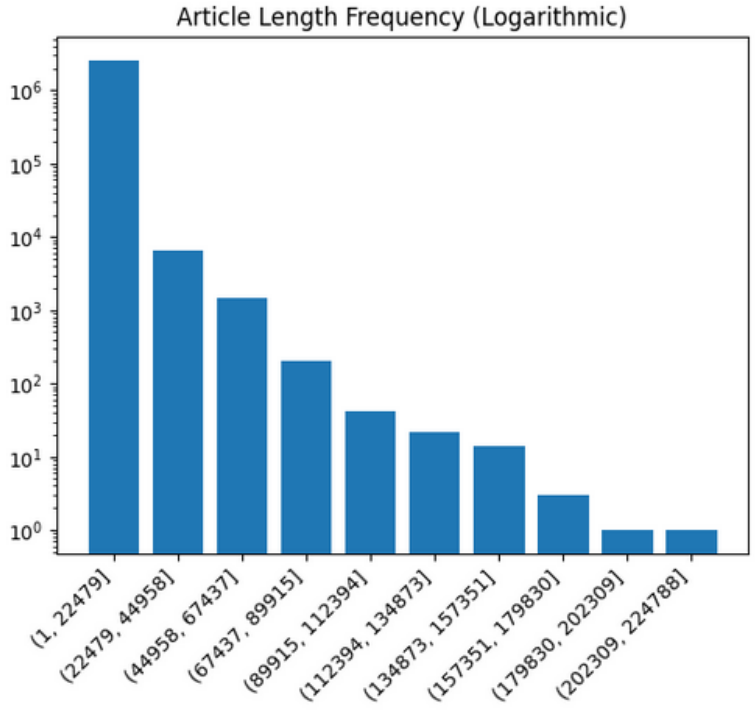

Figure 3 Frequency of articles by amount of characters. (Logarithmic scaling)

The median text length is different across each news sites, with the New York Times, CNN, Vox and The New Yorker being the longest while Axios, Wired, Reuters and TMZ being the shortest. It's important to note that the length of articles are subject to large variances, the articles collected come from a large variety of sources that are not

strictly political news and different journalist may have different methods of writing. Vox magazine and Vice magazine are both publications that contains many articles detailing global politics but also contains articles on opinions on pop culture.

Out of the 26 news sites, 10 of these sites focus more on pop culture and technologies rather than politics and were excluded. (Business Insider, Gizmodo, Hyperallergic, Mashable, People, Refinery 29, TMZ, TechChrunch, The Verge, Wired) Axios is also excluded because the structure of the publication summarizes news into short lists and sentences and is not suitable for the scope of the project.

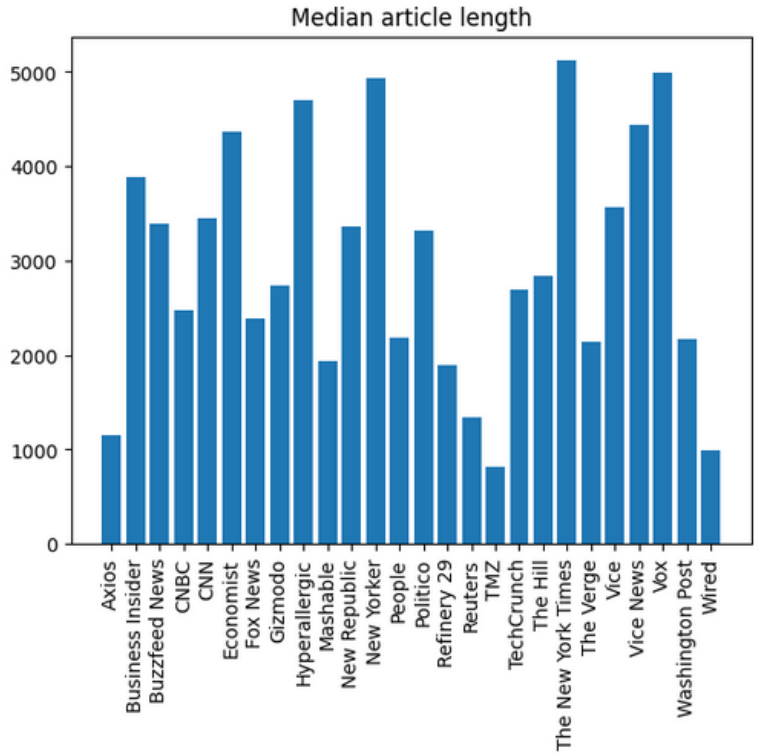

Figure 4 Median Article length by Publication.

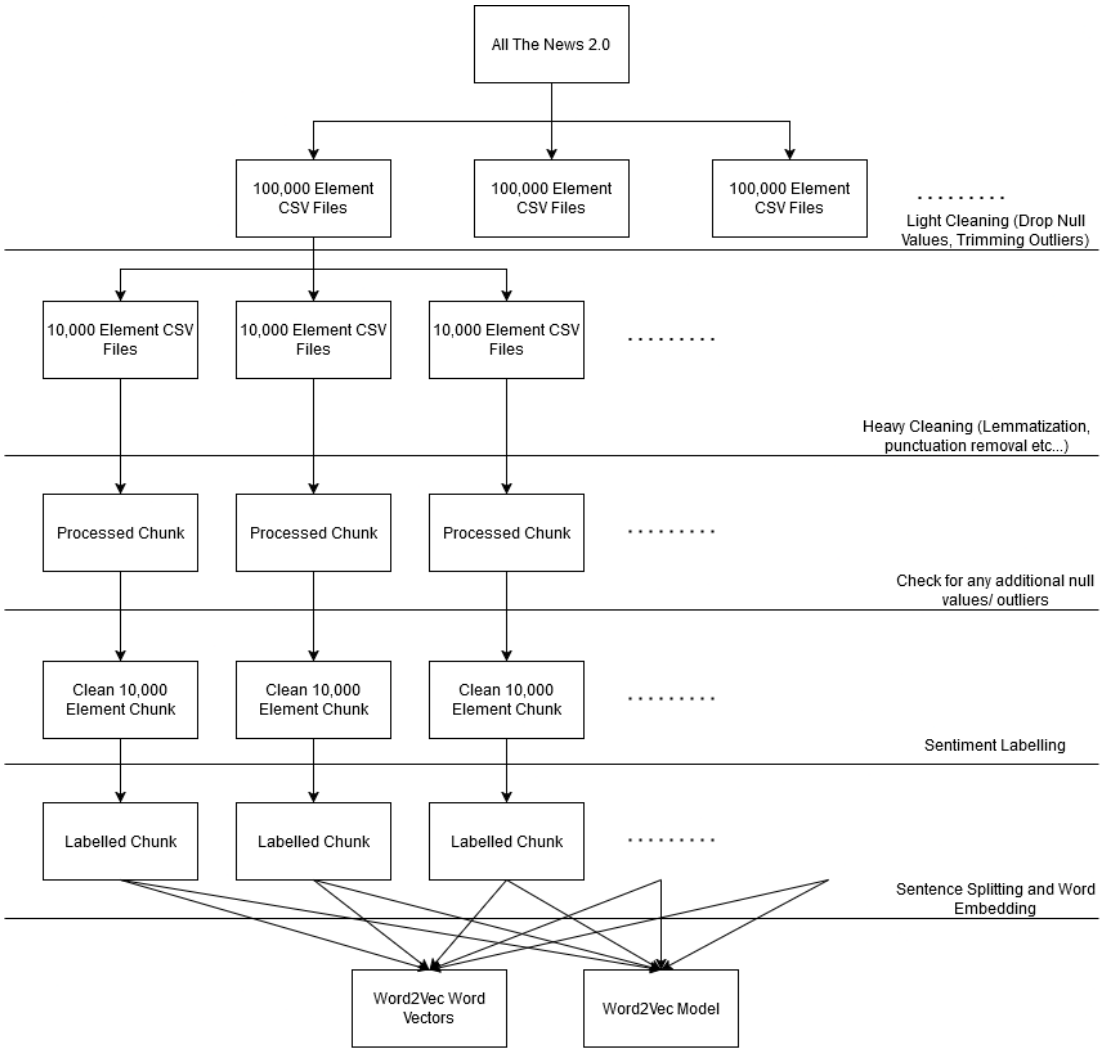

Figure 5 Data Cleaning process.

Data Cleaning

In order to compensate for computing limitations, the data was split into 100,000 sized chunks before processing lighter data cleaning processes like finding null data, then split into 10,000 sized chunks for deeper text cleaning. The data was reduced to only the year, article text and publication. Then, null values within the publication and article text were dropped. If the text length was less than the 1% quartile of its current chunk, it would be considered too short for use and removed. 1% quartile text length is a tiny amount of the data and most chunks have a 1% quartile text length between 200-250 characters. The text is then processed for more rigorous

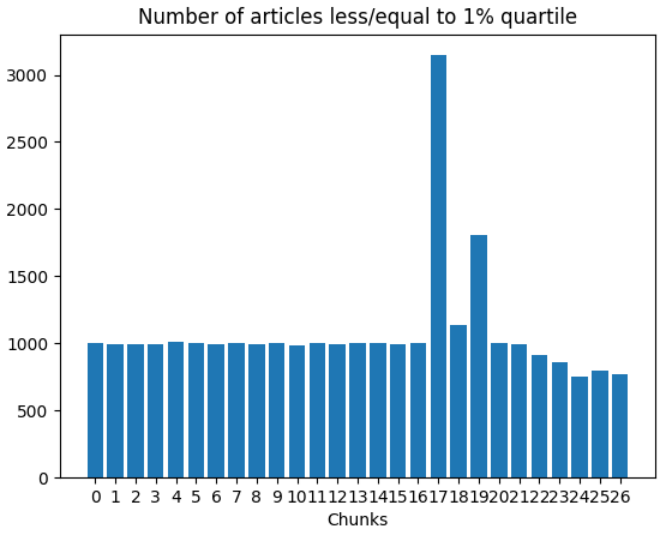

Figure 6 Amount of articles less than 1% quartile in text length.

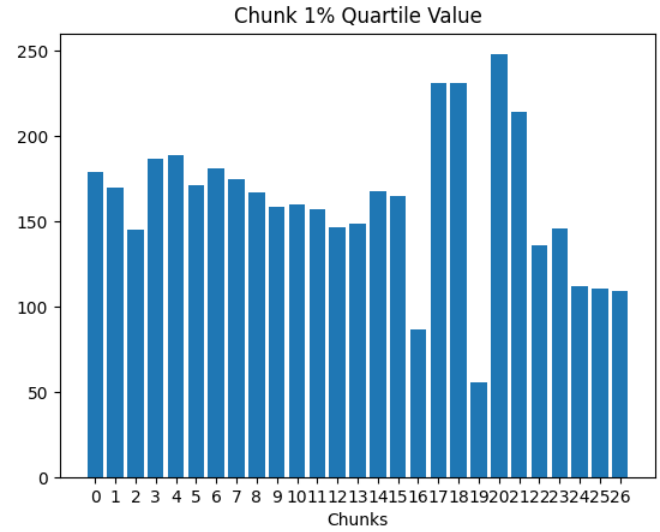

Figure 7 1% quartile character length for each chunk

data cleaning. The individual 100,000 sized chunks are split into 10,000 sized chunks to overcome memory limitations and goes through these removal processes in order:

1. Byte Order Mark (BOM, \\ufeff) removal
2. Duplicate word removal
3. Hyperlink removal

4. Expanding Contractions
5. Removing Non-Ascii/Non-Latin characters
6. Removing Punctuations
7. Convert to Parts of Speech (POS) Tags
8. Lemmatization
9. Stop Word Removal
10. Transform to lowercase
11. Remove POS tags
12. Split into individual sentences

Most methods requires the use of regular expressions (Regex, a form of string matching language) on some level and in total required more than 13 hours of computing to completely clean out the data.

BOM Removal

The first method is removing the Byte Order Mark (BOM) which is a character that signals how Unicode should handle the text either in Big-Endian or Little-Endian. By externally marking this, bytes and characters are properly rendered. This is an invisible character to rendered text and is usually handled but can be annoying for the processing of raw text. (Unicode, 2024)

Duplicate word removal

Duplicated words are removed via the Regex function: re.sub(”\\b(\w+)(\s+\\1)+\\b”,”\\1”,str) which replaces any instance of two words (”data data”, ”sing-sing”, ”waka waka”) into one word (”data”, ”sing”, ”waka”) this can get rid of words in languages that rely on reduplication to modify meanings

like Austronesian Languages or some proper nouns but operating on mostly English articles, this is not as much of an issue.

Hyperlink Removal

Hyperlinks are addresses to websites and may distract from the training and labelling processes. The following Regex is used to detect potential hyperlinks and remove them: ”\S+\.\S+\/?\w+.”. This also detects words that have .net or .org in them which can either be potential website links or organization names that use the styling of domain names for aesthetic/branding reasons.

Expanding Contractions

Contractions are expanded using Regex substitution. The table below details common contractions in the English language and it’s replacement. ”’s” is not

Original Word	Expansion
”won’t”	”will not”
”can’t”	”can not”
”-n’t”	” not”
”-’re”	” are”
”-’s”	””
”-’d”	” would”
”-’ll”	” will”
”-’t”	” not”
”-’ve”	” have”
”-’m”	” am”

expanded into any word because it two major ways of expanding based on it’s syntactic part of speech, either as ”’s” to denote possession of a noun or as a truncation of is. This project will not differentiate between the two and instead removes the ”’s” entirely under the assumption that most occurrences of ”’s” as ”is” appears before a verb (most cases of verbs followed by ”’s” end in the easily identifiable suffix ”-ing”) or adverb (like here, there, up, yesterday, most, etc...) that can easily identified. The possessive

"'s" also frequently appears before a noun. Therefore, the project assumes there to have no significant reason to expand "'s" instead of deleting it.

Removal of Non-Ascii Characters

The corpus is mostly in English. Any Non-Ascii character is removed to ensure that we are working in English. Individual characters instead of words are removed as most of these words are proper nouns like names and places in places that don't use English as a first language, the word is still identified and preserved as a proper nouns during POS-tagging. The Regex used is "[^\x00-\x7F]", which checks for all non-Ascii letters.

Removal of Punctuations and Numbers

Punctuations are used to denote a modification to a word or sentence in English but misplaced punctuations can be a distraction to the analysis, especially numbers where there sentiment value requires more context than the surrounding text. Due to this, it is often best to remove as much punctuation and numbers as possible. This project uses a multi-step Regex substitution to eliminate punctuations (" " Indicates Regex matching string used):

1. Eliminate decimal full stops (1.2, 3.14,...) "\d+\.\d+"
2. Numeral suffixes (-th, -k, -M,...) "(\d+\w+)"
3. Duplicated !, ? or . (!!,...) "\!{2,}","\?{2,}","\.{2,}"
4. Hyphens "-"
5. Every non-alphabet except !,?,. "[^a-zA-Z\s\.\!\?]"

The project still retains at least one ! ? or . as to split the individual sentences later.

Part Of Speech(POS) Tagging

POS tags are used to denote the syntactic category such as verbs, nouns and adverbs along with sub-categories like proper nouns. POS tags originated from the expansion of the work of Noam Chomsky's Syntactic Structures (Chomsky, 1957) and have been in wide use for NLP methods due to it's logical construction of sentences. This step is important for lemmatization as the lemmatized word is dependent on

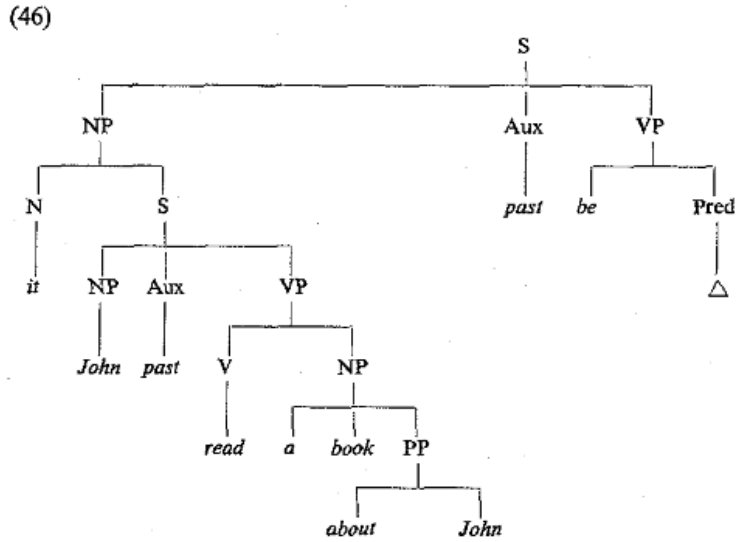

Figure 8 Example of a syntax tree deconstructing a sentence into it's POS categories (Chomsky, 1970)

context it has on the sentence. The sentence is first tokenized with WordTokenizer in NLTK, then converted into a tuple that represents both the word and it's POS tag using NLTK.

Lemmatization

Lemmatization is the act of converting a word into it's root word. This differs from stemming which removes the suffixes and prefixes. By converting a word into it's root, the vocabulary of a NLP model is reduced and any extra meaning that could be carried within the word through it's suffixes and prefixes is removed leaving only

analysis of the root word. However, not all words should be stemmed/lemmatized as some non-root words may contain crucial information being in a certain POS category. As such the POS tag is required for this section.

Stop-word Removal

Stop-words are common words used within a language such as ”I”, ”You”, ”There”, ”Is”, ”Will”, ”Most” and so on. These words are largely uniformative to textual analysis and can often be removed. Important to note that the stop-words used excluded negations and quantifiers like ”not”, ”some” and ”most” as they are important in understanding the sentiment of a sentence. These stop words are excluded from removal. From the NLTK stop word list, they are depicted in the figure below. Articles that contain less than 25 words (outside of stop-words) are removed as they

```
["not","nor","no","few","some","more","most","all","can","will","don","don't","should","should've","now","ain","aren",\
 "aren't","couldn","couldn't","didn","didn't","doesn","doesn't","hadn","hadn't","hasn","hasn't","haven","haven't","isn",\
 "isn't","mightn","mightn't","mustn","mustn't","needn","needn't","shan","shan't","shouldn","shouldn't","wasn","wasn't",\
 "weren","weren't","won","won't","wouldn","wouldn't"]
```

Figure 9 Stop words that were excluded from removal.

may contain data that is not useful enough for analysis or are null after stop word removal.

Lowercase and POS tag removal

After the text is converted to lowercase, the pos tags are removed as the labelling method used (TextBlob) does not rely on POS tagging.

Sentence Splitting

Using the remaining !,? and . Each article is split into individual sentences to prepare the data for labelling and word embedding. A number is assigned to each sentence as to indicate which article it originated from.

After this second round of cleaning, 815,963 articles were removed which constitutes 30.345% of the original data.

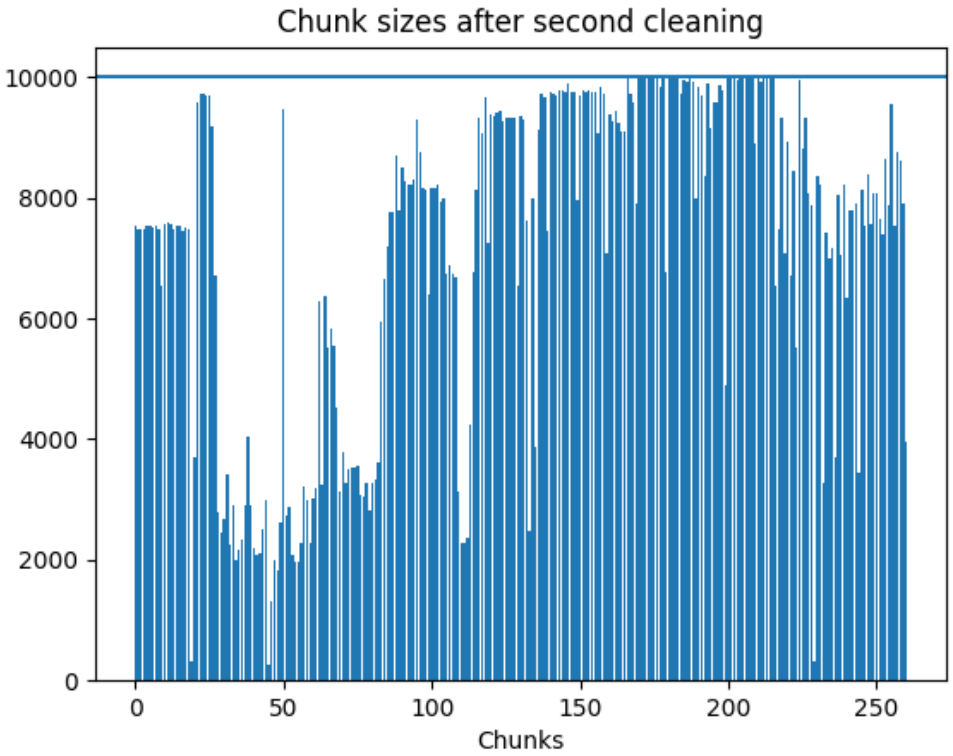

Figure 10 Chunk sizes after cleaning with line indicating the original 10,000 articles per second order chunk.

Rules-based Sentiment Analysis

Word Frequencies

Here are the most frequent nouns, verbs and adjectives present within the data: Much of the news focused around the Republican Party of the United States and

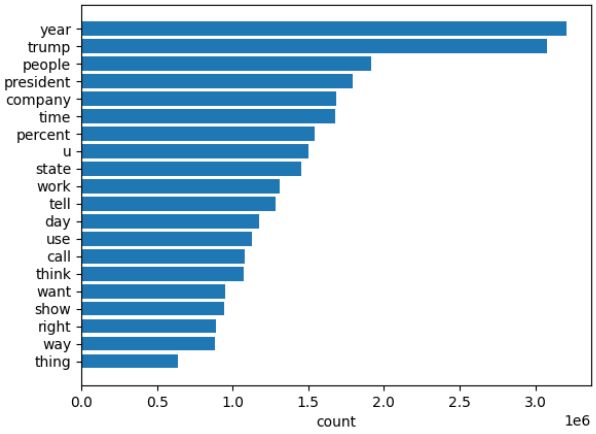

Figure 11 Frequently used nouns within the AllTheNews 2.0 Dataset. (U represents the abbreviation for United States of America, U.S. which was erroneously truncated to u during data cleaning.

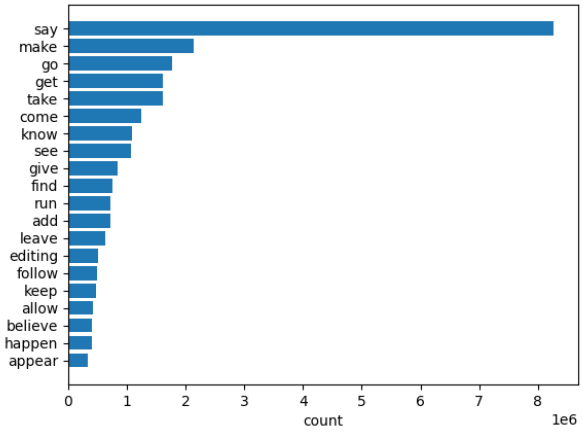

Figure 12 Frequently used verbs within the AllTheNews 2.0 Dataset..

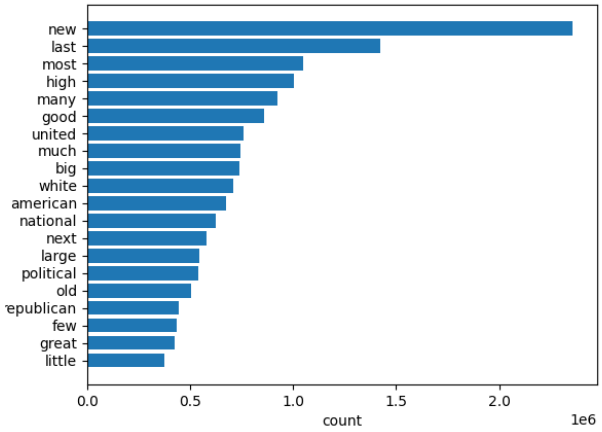

Figure 13 Frequently used adjectives within the AllTheNews 2.0 Dataset.

particular interest was garnered around (as of writing) President-Elect Donald J. Trump and his family with his surname appearing 2nd in most frequent nouns. A lot of the adjectives used are modifiers like high, most, great and big, some of which can be tagged as having positive polarity.

Text Labelling

The data is labelled using TextBlob. TextBlob does not require POS Tags in order to classify sentences which is why the POS tags are removed and the data is rewritten to contain individual sentences rather than paragraphs. The subjectivity score and objectivity score of the sentence is stored along with the original article index. Any empty fields that may have been produced from the sentence expansion are removed.

From the corpus, the polarity scores are labelled in the range between -1 to 1, where 0 is neutral, -1 is highly negative and 1 is positive, while the subjectivity score is labelled between 0 to 1, where 0 is most objective and 1 is most subjective. From the text-blob labelling, the sum sentiment score in each sentence and total sentence count per publication is collected and it's average is calculated by dividing the sum score with the total sentence count. Any objective sentences (that is subjective score == 0) is removed to emphasize the effects that non-objective sentences have. Mean polarity

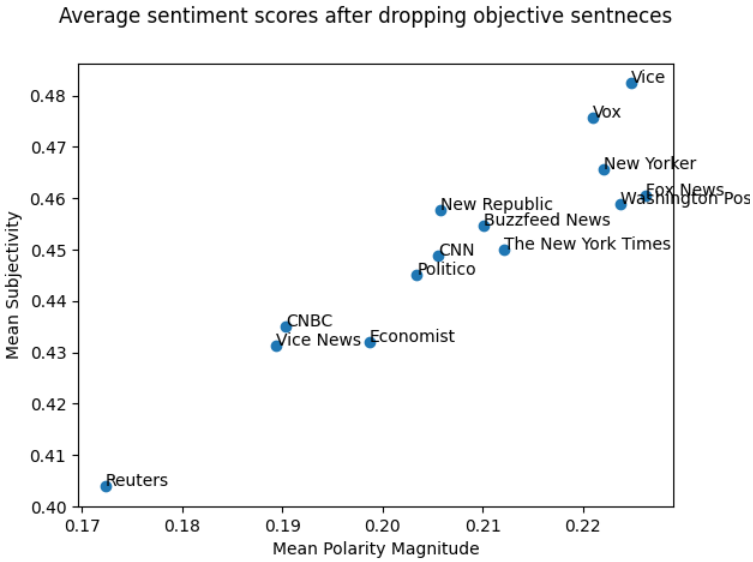

Figure 14 Average sentiment score per publication after accounting for objective sentences. The strong correlation between polarity magnitude and subjectivity scores is expected

magnitude, is also collected by making all polarity positive.

$$N = \text{sentence\_count} \tag{1}$$

$$\text{average\_polarity} = \frac{\sum_{i=0}^{N} \text{sentence\_polarity}_i}{N} \tag{2}$$

$$\text{average\_subjectivity} = \frac{\sum_{i=0}^{N} \text{sentence\_subjectivity}_i}{N} \tag{3}$$

$$\text{average\_polarity\_magnitude} = \frac{\sum_{i=0}^{N} \mid \text{sentence\_polarity}_i \mid}{N} \tag{4}$$

Reuters is noted to have the most objective news as it is also most neutral sounding news outlet. The second most objective publication is Vice News which is stark difference to Vice magazine articles which have high subjectivity scores. Publications like Vice, Vox and The Washington Post have higher scores may be due to them having more opinion pieces and pop cultural news rather than focusing on solely politics. The Spearman's correlation coefficient between mean polarity magnitude and subjectivity is a strong $r = 0.93263013$ which is expected given that objectivity is measured by how neutral a sentence is. Analysing the polarity magnitude, most news publications maintain less than 0.25 in terms of polarity scores which means that most sentences in news articles maintain more neutral language.

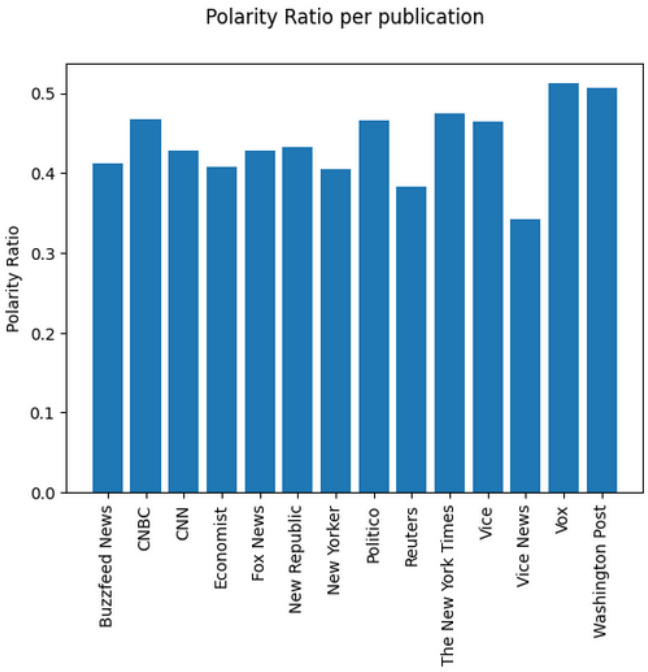

Figure 15 Polarity ratios by publication.

Analyzing the ratio between mean polarity scores and polarity magnitude, the mean polarity is only half of the polarity magnitude and are positive. This means that most publications between 2016 and 2020 have a positive bias in their sentences and are on average around 44% more positive. This may be explained by the usage of quantitative adjectives like high or great which can be tagged as positive. Polarity ratio is given by the following equation:

$$\text{polarity\_ratio} = \frac{\text{average\_polarity}}{\text{average\_polarity\_magnitude}} \tag{5}$$

Media Bias Labelling

The media bias labels are manually annotated using AllSides (*AllSides*, n.d.) (Accessed 11/1/2025), a service that quantifies media bias as left(negative)/center(zero)/right(positive) leaning based on analysis of media publications and community feedback. The labels are applied to the publications and squared to emphasize measure the degree of bias they have. Publications containing more than one section within AllSides are averaged out. (Such as CNN, which has

scores for CNN Opinions, CNN Business and CNN Digital are averaged out). Vice news and Vice magazine are treated as a singular entity in AllSides and thus have the same media bias score applied. Most of the publications outside of Fox News are labelled as either center or left leaning according to AllSides

When plotted onto a scatter plot there is a clear correlation between sentiment scores

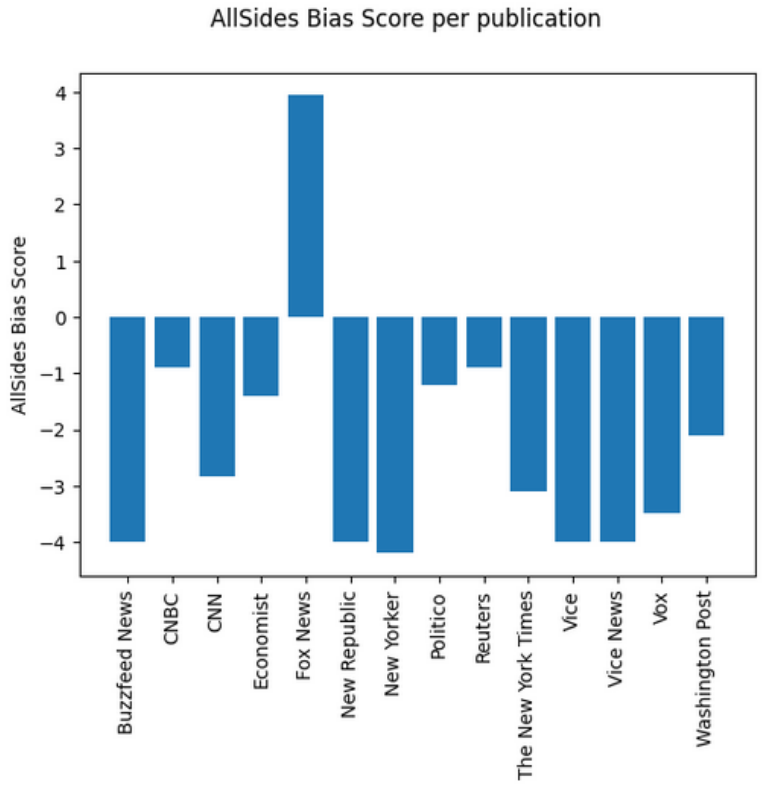

Figure 16 Bias scores provided by AllSides. (Negative indicates left leaning; Positive indicates right leaning;)

and media bias scores. The higher the squared media score, the more subjective and polarizing the journalism is, outside from the outlier of Vice News (As it shares the same bias label as the more opinionated Vice Magazine) and the extremely objective and neutral journalism of Reuters.

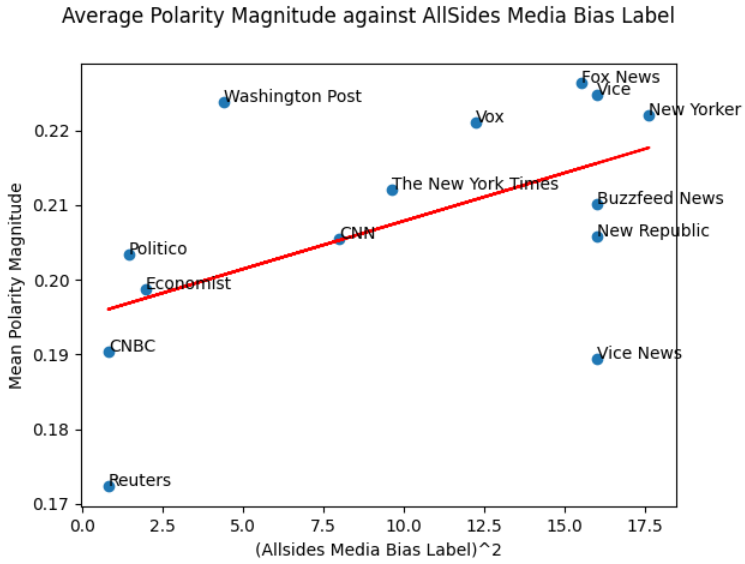

Figure 17 Average polarity magnitude score per publication against AllSides Media Bias label. Regression line provided ($r = 0.53845958$)

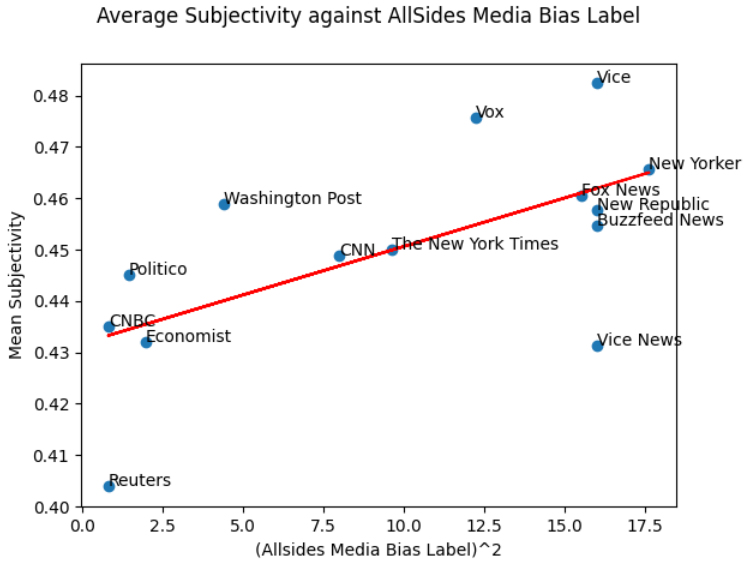

Figure 18 Average subjectivity score per publication against AllSides Media Bias label. Regression line provided ($r = 0.6250412$)

Word Embedding

Words are embedded using a simple Word2Vec model trained using the All-The-News 2.0 data using the GenSim library. Word2Vec is a duo of simple machine learning models both comprising of one hidden layer, continuous bag-of-

words (CBOW) and skip-gram. By default, gensim uses CBOW with the use of the sum of word vectors during training. The model produces vectors of size 300 and takes account of 4 surrounding words when training on a specific word (window size 5). The model also ignores words that appear less than 3 times throughout the corpus. Due to memory limitations the model cannot train on the whole corpus at once, but rather must train using the 10,000 sized chunks. The total training epochs is 1. All the word vectors and model files are saved for future use. The embeddings are visualized

```
model_file = Path(f"./word2vec.model")
if model_file.is_file():
    model = Word2Vec.load("word2vec.model")

for file in file_paths:
    print(file)
    df = pd.read_csv(file)
    sentences = df['sentences'].progress_apply(word_tokenize)
    if model == None:
        model = Word2Vec(sentences = sentences, vector_size=300, window=5, min_count=3, workers=2)
    else:
        model.build_vocab(sentences,update=True)
        model.train(sentences,total_examples=1,epochs = 1)
    model.save("word2vec.model")
```

Figure 19 Training algorithm.

using t-distributed stochastic neighbor embedding (tsne), a dimensionality reduction technique that maps non-linear data onto 3 dimensional/2 dimensional maps. (Belkina et al., 2019) A 2 dimensional map is produced using tsne. A vast majority of words are clustered in a large blob with smaller oblong clusters. There are several smaller clusters that indicate rarer but highly interconnected words. There are many words that fall between clusters possibly due to these words appearing in a large amount and variety of sentences that can fall in either clusters. A 3d mapping may reveal more detailed clusters but due to limitation within this thesis, cannot be made.

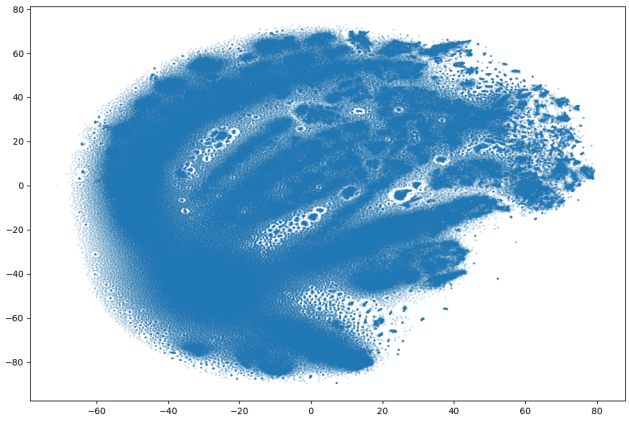

Figure 20 2d mapping of the Word2Vec embedding results.

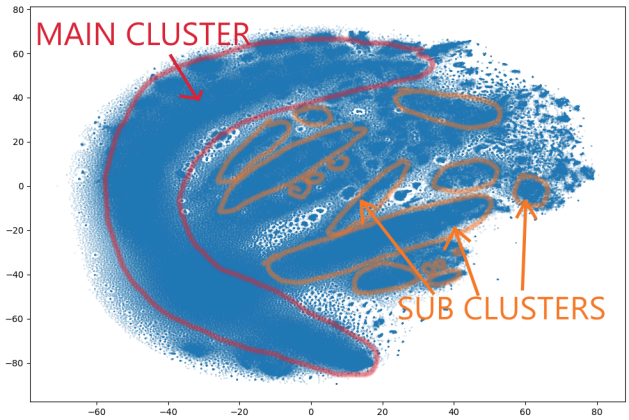

Figure 21 Highlighted clusters. There is a clear main cluster that wraps around the sub clusters, much of the words fall into several of these clusters. Only some of the more clearer sub clusters are highlighted. A significant amount of words also fall in between clusters possibly due to the variety of the sentences that feature such words.

References

Allsides. (n.d.). Retrieved 21-12-2024, from `https://www.allsides.com`

Belkina, A. C., Ciccolella, C. O., Anno, R., Halpert, R., Spidlen, J., & Snyder-Cappione, J. E. (2019). Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature communications*, *10*(1), 5415.

Chomsky, N. (1957). *Syntactic structures*. Mouton de Gruyter.

Chomsky, N. (1970). Remarks on nominalization. *Readings in English transformational grammar*.

Thompson, A. (n.d.). *All the news 2.0.* Retrieved 21-12-2024, from https://components.one/datasets/all-the-news-2-news-articles-dataset/

Unicode. (2024). *The unicode standard version 16.0 core specification.* Retrieved 10-1-2024, from https://www.unicode.org/versions/Unicode16.0.0/core-spec/