Text Analysis of News Articles

By:

Bayat Vahabodin

Contents

Introduction

Text Analysis in Natural Language Processing refers to the process of analyzing and understanding natural language text data using computational techniques. This involves a range of tasks such as tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, topic modeling, and more.

Text analysis of news articles involves using natural language processing techniques to extract useful information and insights from news articles. News articles can provide valuable insights on current events, trends, and opinions. Text analysis of news articles can be used in various fields such as finance, politics, and marketing.

Data Preparation

We started by importing the necessary libraries, including pandas, re, string, and nltk. We then loaded the news articles dataset, which was in the form of a CSV file, into a pandas DataFrame using the read_csv() function. We then used the info() function to display information about the DataFrame, such as the number of rows, columns, and data types.

Next, we defined a preprocessing function that takes in text as an input and performs several preprocessing steps to clean and prepare the text for analysis. The function starts by converting the text to lowercase using the lower() function. It then removes non-alphabetic words using a regular expression pattern (\b[^A-Za-z]+\b) with the help of the re library. The pattern searches for one or more characters that are not in the range of A to Z or a to z at the beginning or end of a word and removes them. The function then removes all punctuation marks using the translate() function from the string library.

Finally, the function removes stopwords, which are common words that do not add much meaning to the text, using the set of stopwords from the nltk library. The words that are not in the stopword set are then joined back into a single string using the join() function.

In last, we applied the preprocessing function to the content column of the DataFrame using the apply() function and stored the cleaned text in a new column called 'content'.

Tasks

The following step adheres to the structure of the assignment.

1- The first 10 lines of the tokenizer's output for the whole corpus.

```
['WASHINGTON', '—', 'Congressional', 'Republicans', 'have', 'a', 'ne
w', 'fear', 'when', 'it', 'comes', 'to', 'their', 'health', 'care',
'lawsuit', 'against', 'the', 'Obama', 'administration', ':', 'They',
'might', 'win', '.', 'The', 'incoming', 'Trump', 'administration', 'c
ould', 'choose', 'to', 'no', 'longer', 'defend', 'the', 'executive',
'branch', 'against', 'the', 'suit', ',', 'which', 'challenges', 'th
e', 'administration', '’', 's', 'authority', 'to', 'spend', 'billion
s', 'of', 'dollars', 'on', 'health', 'insurance', 'subsidies', 'for',
'and', 'Americans', ',', 'handing', 'House', 'Republicans', 'a', 'bi
g', 'victory', 'on', 'issues', '.', 'But', 'a', 'sudden', 'loss', 'o
f', 'the', 'disputed', 'subsidies', 'could', 'conceivably', 'cause',
'the', 'health', 'care', 'program', 'to', 'implode', ',', 'leaving',
'millions', 'of', 'people', 'without', 'access', 'to', 'health', 'ins
urance', 'before', 'Republicans', 'have', 'prepared', 'a', 'replaceme
nt', '.', 'That', 'could', 'lead', 'to', 'chaos', 'in', 'the', 'insur
ance', 'market', 'and', 'spur', 'a', 'political', 'backlash', 'just',
'as', 'Republicans', 'gain', 'full', 'control', 'of', 'the', 'governm
ent', '.', 'To', 'stave', 'off', 'that', 'outcome', ',', 'Republican
s', 'could', 'find', 'themselves', 'in', 'the', 'awkward', 'positio
```

2- The total number of tokens and types (unique tokens) in the corpus.

Total Tokens: 38220484

Unique Tokens: 227034

3- The type/token ratio for the corpus.

Type/token ratio: 0.0059

4- A list of the top 3 most frequent tokens, along with their frequencies.

Rank	Token	Frequency
1	,	1,859,063
2	The	1662,375
3	.	1,455,761

5- The number of tokens that appeared only once in the corpus.

Number of tokens that appeared only once: 89,933

6- A list of the top 3 most frequent words (excluding punctuation and other symbols), along with their frequencies.

Rank	Token	Frequency
1	the	10,519
2	a	5,491
3	to	4,636

7- The lexical diversity (type/token ratio) when using only words.

Lexical diversity: 0.0056

8- A list of the top 3 most frequent words (excluding stopwords and punctuation), along with their frequencies.

Rank	Token	Frequency
1	Said	208,127
2	trump	149,992
3	mr	86,545

9- The lexical density (type/token ratio when using only word tokens without stopwords)

Lexical diversity: 0.0096

10- A list of the most frequent 3 bigrams (excluding stopwords and punctuation) and their frequencies

Rank	Bigram	Frequency
1	mr, trump	343
2	united state	154
3	new york	134