Natural Language Processing Critical Essay
Marissa Beaty
December 5, 2022

Background

To complete this project, I utilized a variety of resources, including: datasets taken from the Kaggle library, one containing “True” news articles and one containing “Fake" news articles; code from in class activities and tasks from the Natural Language Processing notebooks of weeks 2 and 5; research conducted in the NLTK Library database; and a collection of pre-set python packages, such as Keras Sequential, Dense, and Layout; Gensim KeyedVectors; SKLearn train_test_split, CountVectorizer, and TfidfVectorizer; Pandas, and more. With these resources, I have adapted and written all code necessary to topic model and classify the “True” and “Fake” news articles.

Project Aims

My final Natural Language Processing class project involved using two pre-made datasets from the Kaggle library to showcase my understanding of topic modelling and classification. The first dataset is a collection of “True” news articles; the second dataset is a collection of “Fake” news articles. Each dataset contains the article title, text, subject, and date of publication. As I am most interested in the article Titles, I have elected to use that data to conduct the topic modelling and classification research.

With this data, I seek to do two things. First, identify common topics and words associated with the “True” and the “Fake” articles to investigate in what ways the titles between the two are similar or different. Beyond this, are there specific words readers can look for to make better choices on what news articles they read.

Following, I will create an LSTM classifier that can determine an articles “True” or “Fake” status given a sample of the article titles in my dataset. The purpose of this classification is to let a computer sift through the true and fake articles for you.

Methods

Since I will be working with two NLP concepts (topic modelling and classification), I will split my methods into two sections. Before delving into the NLP topics, though, I had to first prepare the data to be manipulated and used. I will describe that process now and then discuss what I did once the data was processed.

First and foremost, I imported a variety of necessary packages all at once. Later, I imported additional packages as they became necessary, but by importing the majority at the top, I saved processing time and cleaned up the look of the notebook. Once completed, I turned my datasets into data frames, one containing the “True” articles and the other containing the “Fake” articles. The original dataset was quite large, with over 45,000 rows. To make this easier to work with, I took a random sample of 5% of the data to use for this project. Though this step is not preferred, utilizing more than 5% of the dataset caused significant delays in the testing

process. Once completed, I created a new column in each dataset titled "label." This column holds a 0 for the "Fake" articles, and a 1 for the "True" articles. Finally, I combined the “True” and “Fake” datasets together, saved that file as a .csv to my local folder, and read that file back into jupyter for later use.

Once this process is complete, it is time to move onto cleaning up and tokenizing the data for topic modelling. This process required three steps. First, I replaced all instances of “U.S.” to “unitedstates.” After reviewing the data both before and after tokenizing, I noticed instances of "U.S." were removed from the dataset after being cleaned. As the dataset is all U.S. based news articles, I deemed it necessary to adjust the dataset to ensure this term is included when modelling the data. Second, I remove all special characters from the dataset using Regex. Third, I check if any words are in my stop words list before adding the words to a final “go_words” list. The stop words were imported from the NLTK corpus. To simplify this process, I defined a function that passes text through all three tasks and returns a list of cleaned tokens. I have elected not to stem my terms primarily because doing so caused too many important terms to lose their meaning. As I am also interested in the frequency of terms, I also elected keep all duplicate terms.

Once my tokenizer has been defined, I move onto topic modelling the titles. As I want to look at the “Fake” article titles comparatively to the “True” article titles, I conduct topic modelling on each separately. The process for topic modelling was adapted from our Natural Language Processing Lectures from Week 2. First, I utilized a tfidf vectorizer (with my vectorizer defined) to pull out the vocabulary and shape of the dataset. This also prepared the tfidf dataframe to run through the TruncatedSVD function. To keep the process running, I elected to only pull 5 topics. After running through the TruncatedSVD function, I displayed a sample of topics and topic weights to ensure the process is working. Finally, I pulled the top 5 topics, and printed 20 terms within each topic. This process is repeated for both the “Fake” news titles dataset and the “True” news titles dataset.

Once this is complete, I move onto the second part of this project: training an LSTM classifier on the total dataset. This process was adapted from the Natural Language Processing Week 5.2 Lecture materials. Since I am training a classifier on all data, I used the combined_news_dataset I created at the beginning of the project. To prepare this data for the classifier, I first reduced the dataframe to only include the article titles and labels. I then uploaded the “GoogleNews” vector’s file. I also reverted to the tokenizer and vectorizer in the Week 5.2 lecture to avoid conflicts created using my own vectorizer defined earlier. I also defined the pad_trunc function from lecture 5.2 to ensure all my vectors are the same length.

Having imported these functions, I ran my new dataset through the tokenizer and vectorizer and split the dataset into training and test sets before padding them into three dimensions. I then defined my batch size, number of neurons, and the number of epochs. I have specifically reduced the number of epochs from 5 to 3 as previous practice training on other data had a diminishing returns of training accuracy above 3 epochs. Next, I used tensorflow Sequential to

set up the model before running my data through the training model. My last step is to save the model and model weights so it can be used with other data.

Results

I trained my topic model to produce 20 terms in the top 5 topics for the "Fake" news titles and "True" news titles. Within the "Fake" topics, there is a similar trend of terms contained within each topic, including, but not limited to names of former presidents and presidential candidates (Clinton, Trump, Obama), terms of foreign governments and leaders (Saudi, Brexit, China, Korea, Syria), and finally a significant number of terms around the public facing aspects of a campaign (News, Speech, Campaign, Poll, Media, Watch).

The "True" title model had very similar results, the difference being the addition of another category: numbers. What is not present in the "Fake" titles' topics, and is in the "True" titles topics is exact numbering, such as 100K, 108, 02, 10th, etc.

My classifier was also quite successful. During the three epochs, my model went from an accuracy of 76.38% to 98.6%, and a loss of 0.6059 to a loss of 0.2234.

Discussion

The results of my topic modelling project are quite interesting, particularly since there is not a drastic difference between the "Fake" titles topics and terms and the "True" titles topics and terms. In fact, the only significant difference found between the two is the use of numbers in the "True" titles. I believe this presents an interesting discussion into the ease of reading "Fake" titles thinking they are "True." It also, however, proposes additional research questions: if the terms within the titles themselves are not so different, how can we (humans) determine the difference between what is a "True" and "Fake" title? Is there another way we can identify the difference, such as is the word order different? Or perhaps the title length varies? It would be interesting to look further into these questions, especially in coordination with other research departments, particularly political science, or psychology.

As for the classification process, I am pleased with the results of my classifying model. Though it does not have perfect accuracy, I was able to yield a near 99% accuracy on my dataset. I think the 1% could be a result of inconsistencies derived during the tokenizing and vectorizing process. When using the tokenizer and vectorizer function as defined in lecture 5.2, the data did not get as clean as preferred. Therefore, I defined my own tokenizer when conducting the topic modelling. Since I revert to the Lecture 5.2 tokenizer for the classification process, I believe it is possible this caused some faults in the data that limited the training models accuracy.

Overall, I completed the project as I intended to, however, had I had more time, I would like to have delved further into each piece of the project. For topic modelling, I would have like to have asked more questions on the data. There are also a few cleaning errors I would like to have fixed. I also would have been interested in graphing the results of the data comparatively, to visualize what terms are common and unique within the datasets. Similarly, with the

classification section, I would like to have inputted new data into the model to test the accuracy. I would also have liked to try different models beyond the LSTM to see how the accuracy of a different model compares to that of the LSTM model. As mentioned, though, I am happy with the results and consider it a strong foundation for further research.

Ethical Considerations

The purpose of this project, as described in the "Project Aims" was to successfully topic model and classify the "Fake" and "True" news article titles. What exists beyond the code, however, is the hope that in completing these tasks, I have created an avenue for which readers of news articles can be more conscious of what it is they are reading, and thus, be more critical of it.

I would be remiss, however, to not discuss how this could potentially be used negatively. The ethical consideration of a project like this is, of course, important when looking at how the code is written, but more so in how the results are interpreted. If a model like this were ever used, it would be essential to emphasize the necessity to look further into the articles themselves before drawing conclusions about whether an article is "True" or "Fake." This is especially true when looking at the topic modelling. As there are significant overlap in terms between the "True" and "Fake" articles, to identify an article as true or fake solely on the terms presented in the title could not only be inaccurate but misleading. As like anything involved with Machine Learning, human intervention is necessary to validate accuracy.

References

Bisaillon, C. "Fake and real news dataset: classifying the news." *Kaggle.com*. Available at: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset (Accessed: 5 December 2022).

NLTK Library. "Stop Words." *NLTK Stop Words Search Results.* Available at: https://www.nltk.org/search.html?q=stopwords (Accessed: 5 December 2022).

University of Arts London. "Lecture 2.2 Topic Modelling." *Natural Language Processing*. Available at: http://localhost:8888/notebooks/Documents/GitHub/nlp-22-23/NLP%20Week%202.1-Text%20as%20Numbers-class.ipynb (Accessed: 5 December 2022).

University of Arts London. "Lecture 5.1.2 Sequential Data." *Natural Language Processing*. Available at: http://localhost:8888/notebooks/Documents/GitHub/nlp-22-23/NLP%20Week%205.1.2%20-%20LSTM%20Classification.ipynb (Accessed: 5 December 2022).