Kyle Muncie, Randall Powers

CSE472

Project 2

Abstract

From the last few years, the use of social media has increased resulting into the rise of fake news and their spreading on a large scale. As seen by the widespread impact of the huge beginning of fake news, people are inconsistent in the absence of effective fake news detectors. In this project, we worked on fake news recognition for 2 different datasets. One of those datasets being the train.csv file to train our model, while the other dataset being the test.csv file to test our model. The results are compared with the actual labels of the test.csv file. With the help of selenium, google translate, data analysis, and our model, we were able to effectively deal with this issue. The proposed method is compared with many other approaches on the website called Kaggle, where we are performing against our classmates. This approach has created a model that captures various fake news indicators and classifies the news as false (0), misleading (1), true (2), or unproven (3).

Introduction

Nowadays, people vastly rely on social networking platforms to get knowledge, information and current happenings as it provides the same information in a quite new, fluent, speedier and portable manner. Starting with this topic, we discussed how we were going to approach this issue of misinformation. We understand from some research that misinformation consistently spread from one source should also be the same label (0) as others. Previous work has been done before to try to blacklist these people or websites but the approach we took was to make a model that was to first look at key words based on the data analysis we did and then if

there was no match, we would then scrape the fact check URL and based on the information our model found there, it would then return the correct label. We believe our model is a great way to depict misinformation because it doesn't just scrape a website, it uses prior learning based on the training.csv file. While scholars have often argued that blacklisting might be the best way to prevent misinformation, we argue that our model would be a better fit because it looks at correlations and also scrapes the fact checked URL.

Related Works

The section outlines the recent works in the detection of fake news. Basically, there are 4 main types of fake news detection techniques viz. The first technique, data-oriented approach primarily focusses on psychological and temporal data. The second technique, feature oriented approaches try to find out novel features from the news context or through social context. The third technique, the model oriented one tries to fit some supervised, semi-supervised or unsupervised classifier to predict the genuineness of the news. The last technique, the application oriented one tries to predict the diffusion or intervention of the news. This pre-existing knowledge and the knowledge gained through our own research has led us to our current work where we combine the fake news techniques of data and model to form our current working model.

Model Description

Our model uses pandas for data manipulation, selenium for web scraping, html to text for reading the html of the page, and google translate to be able to translate other languages into English. Based on our research, we wanted to first build our model based on key words we found from the train.csv file and then the claims that did not match, we would then scrape the fact check URL to return the label our model believed would be correct. When we are scraping the

URLs, we would be returning labels based on keywords from the html that we scraped. When our model finds a hit in any of the processes above, it then outputs the label to the csv file which is then submitted to Kaggle.

Experiment

Early experiments we ran in this were with the train.csv file where we used our degree distribution from the first project to get key words based on the label in the csv file. We only kept the key words that unanimously had one label. The other claims that were not solved by this method, we decided to use selenium to get those labels. We tested our model with the train.csv file to see if we got around the same results as the labels that were already there and it had about an 80-90% accuracy. We experimented grabbing various elements on the ASU website when we were scrapping and we finally decided to use the html element since it is in every website no matter what. We then tested if we could compare a string that we had to something we knew was in the ASU website and we were able to find that. Based on that, we believed we should be able to get the correct labels.

Future Works

Future work in our model includes having a better way to grab an html element when using selenium or even using another method in whole. Selenium as a package takes a long time to grab html elements from the fact check URLs.

References

- https://www.selenium.dev/documentation/webdriver/
- https://pypi.org/project/googletrans/
- https://pypi.org/project/html2text/
- https://pandas.pydata.org/pandas-docs/stable/reference/index.html

- https://towardsdatascience.com/managing-infodemics-slowing-the-spread-of-misinformation-b8b74e3e2618