Fake News Content Detection

Han Li (hl565), Leo Tang (lt342)

Description

With the prevalence of fake news content perforating the internet and our increased reliance on social media for information, it has become important to think about how we interact and approach news on the internet. By incorporating machine learning and artificial intelligence, our goal is to discern fake/conspiracy news from trustworthy news sites. Given a text of a news article content, our algorithm will identify whether or not the input contains fake information. This classifier will be accessible through a simple command line prompts and, if time permits, a full fledged web GUI that can also display statistics including how many percent of each website's content contains fake news.

Although the classifier will most likely work best given a long input string, we aim to create a algorithm that can classify news article simply with title or from few key sentences from the news article itself. In addition, being able to display feature statistics such as displaying what word usage most contributes to an article being fake would be useful to the user. We believe that the algorithm should do more than simply give the users answers. In fact, the classifier should be used to help users become more informed about what keywords and tags to look for in an article to verify its authenticity.

Ultimately, our goal is create a highly accurate, in terms of both precision and recall, classifier that can identify fake content based on few key sentences. In order to accomplish this goal we will require myriad of data from both fake/conspiracy articles and trustworthy articles. We believe that getting enough data will be one of the harder aspects of this project. Nonetheless, we have access to few datasets from Kaggle that will provide us with a good starting point for this project.

General Approach

We will first download the relevant datasets from Kaggle instead of parsing ourselves from AP and the Onion. This is to minimize time spent on cleaning and parsing data rather than on training our model. The fake news dataset is located at https://www.kaggle.com/mrisdal/fake-news/data, and contains approximately 13,000 articles containing title and text with various metadata including publishing website. The real news dataset is located at https://www.kaggle.com/uciml/news-aggregator-dataset/data, and contains 400,000 from trustworthy websites. These two combined together 50/50 will make up our overall

dataset, which is around 26,000 to 30,000 articles. We can then preprocess the data with lowercasing all words and stemming.

We then can begin by building a baseline model using logistic regression. We plan on using GLOVE for word embeddings. The model will then take in a couple of features. One feature is the word embedding average of all the words appearing in an article. This will allow us to find correlations between words that might indicate a fake news article. A bag of words model does not model this relationship, and thus probably would not work as well for us. A second feature might be the publishing website, which can be very indicative of fake or real news.

Our alternative design will be a two-layer feedforward neural network, which takes an input of the word embeddings average, and outputs the binary label. We keep taking the average in order to consider all words in the article.

Finally, we plan on allowing the user to input a block of text on the command line, and get a percentage on how likely the article is to be real or fake news.

Evaluation

We plan on splitting the dataset into 80% training, 10% validation, and 10% testing. This is to ensure that we are able to evaluate how good our detector is at fake news detection. We would then simply take the dataset we reserved for testing, and run them through the models. Because we know the existing labels for the testing set, we can calculate precision, recall, and consequently the F1 score. This will be our measure of how good our predictors are. We initially anticipate the result of the logistic regression baseline to be around F1=0.6, which is what our initial research suggests. The alternative design will hopefully reach F1=0.8 after tuning and validation.

Timeline

3/15 - 4/1:

- Download the real and fake news dataset and merge to get 50/50 distribution
- Research more about NLP text featurization on Neural Networks

4/1 - 4/15:

- Implement logistic regression model, evaluate
- Implement neural network model, evaluate
- Begin work on command line GUI

4/15 - 5/1:

- Continue working on command line GUI, if time permits move to web GUI
- Improvements and optimizations for model and approach