CAPSTONE PROPOSAL: FAKE NEWS DETECTION

Domain Background

Recently, news consumers are becoming more and more aware of the **Fake News** problem. Propaganda funded by certain states and wealthy people find their way into people's Facebook feeds directly, and indirectly as they get referenced by other media channels.

It is important to address this issue as it interferes with people being informed for democracy to prevail. Journalism has been keeping the politicians in check by exposing corruption and informing the audience about their actions. But fake news create noise, and confuse and misinform readers.

This issue has been a topic of numerous academic studies including machine learning based study at UCSB. William Yang Wang from UCSB has published his findings on statements made by politicians and political entities in the US(checked by Politifact), and has been able to guess with a much better accuracy than guessing randomly by using CNNs.

Problem and Solution Statement

The problem to solve is simply given an article, can we guess if it's real or fake news.

Guessing if a given article is fake news or relatively reliable is going to take Natural Language Processing techniques combined with Machine Learning. To start with, I'm planning to use reading ease level, sentiment analysis, naive bayes classifier, and CNNs.

My metrics are going to be accuracy, F1 score, and recall of fake news. For the future, it would be great to have a warning for the potentially fake news so I would double check its accuracy.

Anyone should be able to work on the same problem using the same dataset as I do.

Datasets and Inputs

After searching for a good dataset, my top 2 choices so far are Kaggle and George McIntire.

Kaggle: The fake news dataset on the site takes disk space of about 57MB, has 13,000 rows(entries) and 20 columns.
Pros: Pretty neat, many columns(structured information)

Cons: No real news(so you'd need to find your own), only from a span of a month, seems to only include extreme right wing sites

Source: https://www.kaggle.com/mrisdal/fake-news

George McIntire: His dataset includes fake news and real news in 1:1 ratio. The dataset takes a disk space of 31MB, 10,558 rows(entries) and 4 columns.
Pros: Pretty neat, simpler, contains real news, data from both left and right wing sites

Cons: Less information for each entry, the labeling might not be 100% right(when is it ever 100%?), only two options 'fake' or 'real'(no in between)

Source: https://github.com/GeorgeMcIntire/fake_real_news_dataset

Finally, I would like to give a mention to William Yang Wang from UCSB. His dataset does not contain news, but it contains statements made by major politicians and political groups in the US. The statements are rated from 'true', 'half-true', 'barely-true', 'false', down to 'pants-fire', which makes it quite more interesting. And both right-wing and left-wing groups have their share of falsehoods, so the political bias might be less of a factor. However, it doesn't actually have news content; it only has short statements that are typically sensational.

Benchmark Model

George McIntire wrote this on opendatascience.com:

"Out of the 5234 articles left in the other fake news datasets, my model was able to correctly identify 88.2% of them as fake. This is 3.5 percentage points lower than my cross-validated accuracy score, but in my opinion it is pretty decent evaluation of my model."

He doesn't give all the details about the specifics, so it's hard to verify if his claims. One can hope he is not making this up!

Regardless, 80% recall seems like a good benchmark, and I'll look into improving accuracy and F1 scores with different methods I use.

Evaluation Metrics

Since the model is going to classify the data points, I'll use three metrics; accuracy, F_1 score, and recall rate. I don't want to focus on improving only one metric in a biased way, so I went with three metrics that could all be relevant depending on the purpose.

1) Accuracy = (true positive + true negative) / (dataset size)

2) F_1 = 2 * (precision * recall) / (precision + recall)

F_1 score is the harmonic average of precision and recall.

For precision and recall, I'll give the quick example from Wikipedia:

"Suppose a computer program for recognizing dogs in photographs identifies eight dogs in a picture containing 12 dogs and some cats. Of the eight dogs identified, five actually are dogs (true positives), while the rest are cats (false positives). The program's precision is 5/8 while its recall is 5/12."
https://en.wikipedia.org/wiki/Precision_and_recall

3) Recall = (Number of Fake News Identified) / (Fake News Count)

I want to use Accuracy and F_1 score as they're commonly used as metric for classification algorithms. So if someone else works on the same problem, they'll have an easier comparison. And finally, I'd like to use Recall as the data scientist that created the dataset I'm using has stated its finding for the rate of fake news identified out of all the fake news.

Project Design

Currently, I do not have a clear plan on the exact structure of the model. However, I will use the methods I mentioned earlier such as reading ease level, sentiment analysis, naive bayes classifier, and CNNs in different combinations to maximize the metrics.

Along the way, I'll likely try new methods until I reach satisfactory levels for all three of my metrics.