Fake News Detection Using Sentiment Analysis and Metadata

Maniz Shrestha (mashres15@earlham.edu)
Department of Computer Science, Earlham College, Richmond, IN

INTRODUCTION

Fake news is any form of false story or content spread on the internet to influence people's view to gain inimical benefits. Detecting fake news in the digital world has been a significant challenge in overcoming the widespread dissemination of false rumors and biases. Although there has been significant progress in fake news detection, a standard set of solutions is yet to be established. Companies such as Facebook, Twitter and Google are facing challenges in tackling this problem to ensure a platform where people can trust the newsfeed content. This study used a hybrid approach of combining sentiment analysis with the network approach of using metadata to create a fake news classification model. A user-friendly web interface was implemented to enable users to easily query a news source using a URL and determine whether the news is fake.

CONTRIBUTIONS

The following are the major contributions of this research project:

- Fake news detection model using Facebook social media integration.
- Scraping architecture for news content, metadata, and social media analytics.
- A web-based application that takes a URL and classifies the news or article as “fake or reliable.”
- An extension of the Fake News Corpus built using the query result from the web application.

DATASETS

The FakeNewsCorpus[1]
-> 9 Million+ records → **192,926 records**
After cleaning, filtering & random selection

Getting Real about fake News (GRFN)[2]
-> 13,000+ records → **12,345 records**
After cleaning & filtering

DESIGN & IMPLEMENTATION

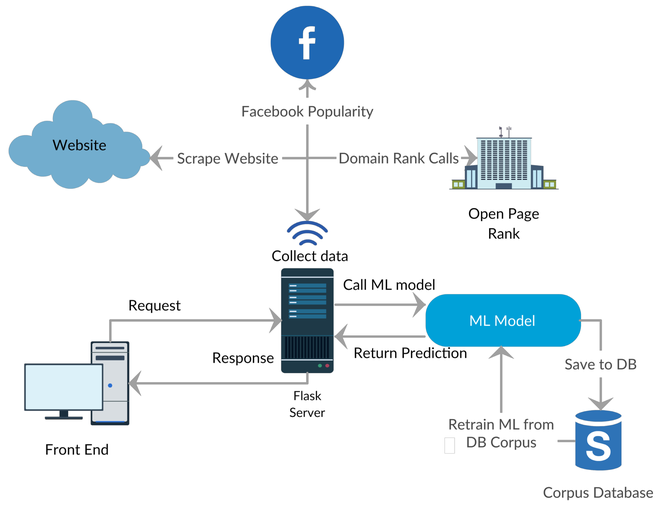

Figure 1: The System Architecture

The front-end is backed by the flask server which collects the news source data, implements the **Random Forest model[3]** and stores the results in the database. The Random Forest model is implemented using **Scikit-Learn[4]** and has four pipelines for feature engineering. The text pipeline (see Figure 2) converts the text content to document vectors, the sentiment pipeline generates the polarity and subjectivity of the text, the numeric pipeline normalizes the numerical data and the hashing pipeline implements feature hashing for other features. All of these features are passed to the Random Forest Classifier for training the model. The trained model was implemented in the web application to classify fake news.

Technology stack: Python, Pandas, Scikit-Learn, Gensim, Textblob, Flask, HTML, CSS, and JavaScript.

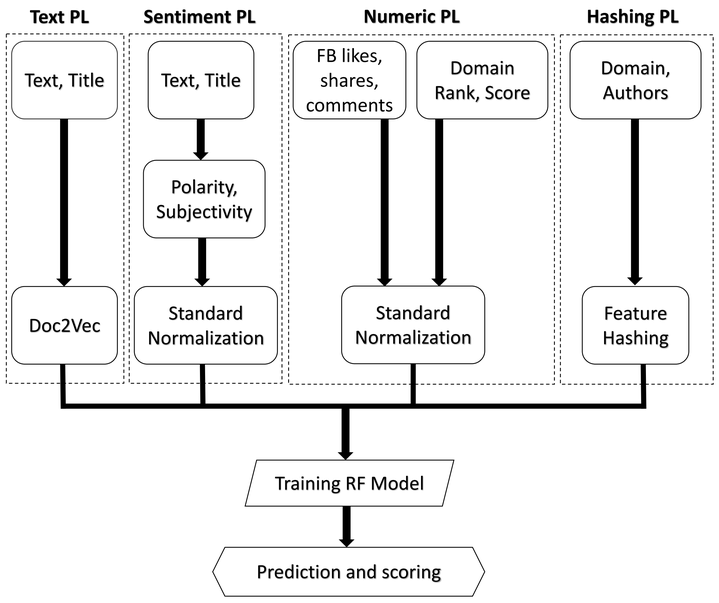

Figure 2: The Machine Learning Pipeline

RESULTS

For the first test, the FakeNewsCorpus was divided as 70% training and 30% testing set. Using this dataset, the model achieved an accuracy of 99.9% and an F1-score of 99.99% (see Table 1):

	Precision	Recall	F1-score	Support
0 (reliable)	1.00	1.00	1.00	29,874
1 (fake)	1.00	1.00	1.00	28,004
Avg/total	1.00	1.00	1.00	57,878

Table 1: Classification Report for FakeNewsCorpus 70-30 test-training

For the second test, the model was trained on the FakeNewsCorpus and tested on the GRFN dataset plus 5,000 reliable news from the FakeNewsCorpus to see how the model performs on a different dataset. The model achieved an accuracy of 82.27% and an F1-score of 90.23% (see Table 2):

	Precision	Recall	F1-score	Support
0 (reliable)	0.70	1.00	0.83	5,000
1 (fake)	1.00	0.83	0.91	12,345
Avg/total	0.91	0.88	0.88	17,345

Table 2: Classification Report - training on FakeNewsCorpus and testing on modified GRFN

CONCLUSION

This study proposed a hybrid approach using linguistic cues and network metadata for fake news detection and reached a high accuracy rate (100% precision, 82% recall, 90% F-score). The web user interface demonstrated the applicability of the model in limiting the rapid spread of unreliable content on the Internet. Areas of further investigation include using fact checking and deep syntax analysis, as well as recommending similar credible articles.

REFERENCES

1. MaciejSzpakowski.2018.FakeNewsCorpus:Adatasetofmillionsofnew sarticles scraped from a curated list of data sources. https://github.com/several27/ FakeNewsCorpus
2. Megan Risdal. 2018. Getting Real about Fake News. https://www.kaggle.com/ mrisdal/fake- news
3. Leo Breiman. 2001. Random forests. Machine learning 45, 1 (2001), 5–32.
4. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cour- napeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python . Journal of Machine Learning Research 12 (2011), 2825–2830.

Acknowledgement: This project was supported by the Department of Computer Science at Earlham College. The author would like to extend sincere gratitude to Dr. Xunfei Jiang, Dr. Charles Peck, Dr. David Barbella and Dr. Ajit Chavan (faculty advisor) for their guidance in the design and evaluation of the project.