Data Provenance Summary

Fake and Real News Dataset
Unknown License
Retrieved {DATE} from:
https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

This data set contains 2 CSV files.

- True.csv (21,417 articles)- composed solely of real news articles split between 2 subjects.
    - World-News, 10,145 articles
    - Politics-News, 11,272 articles
- Fake.csv (23,481)- contains only fake news articles split between several subjects.
    - Government-News, 1,575 articles
    - Middle-east, 778 articles
    - US News, 783 articles
    - Left-news 4,459
    - Politics, 6,841
    - News, 9,050 articles

Data Collection Method

	True.csv	Fake.csv
Source	https://reuters.com	A variety of websites flagged as unreliable by Politifact.
Method	Web scraping (no package specified)	Web scraping (no package specified)
Processing	Cleaned and processed (No specifics, but at least HTML tags removed.)	Cleaned and processed, but spelling errors and punctuation preserved. (No further details specified, but HTML tags were removed.)
Fields	Article Date Article Title Article Type Article Text Label (I did not see this, and added it on my own)	

Related Studies

The dataset at Kaggle cites two articles appearing below.

Article 1

Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.

Excerpt: “In this paper, we introduce a new n-gram model to detect automatically fake contents with a particular focus on fake reviews and fake news. We study and compare 2 different features extraction techniques and 6 machine learning classification techniques.”

As of 28 Aug 2023, I am unable to find a PDF. I will continue looking, but based on each article’s abstract, the approach in this article may be similar to what is described in the second one.

Article 2

Ahmed H, Traore I, Saad S. (2017) “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138).

Excerpt: “We propose in this paper, a fake news detection model that use n-gram analysis and machine learning techniques. We investigate and compare two different features extraction techniques and six different machine classifi- cation techniques. Experimental evaluation yields the best performance using Term Frequency-Inverted Document Frequency (TF-IDF) as feature extraction technique, and Linear Support Vector Machine (LSVM) as a classifier, with an accuracy of 92%.”

Earlier work that informed this article broke language components from articles into 3 groups:

1. Complexity features- features calculating the complexity and readability of the text.
2. Psychological features- features assessing the cognitive processing and personal concerns within articles, such as counts of emotion words and casual words.

3. Stylistic features- traditional stylometric features, such as counts of parts of speech.

The above features fed into an SVM classification model that showed 71% accuracy when judging between fake and real news articles. No additional metrics, such as precision, sensitivity, or specificity were mentioned in the literature review.

Approach and Models

This article outlines a 4-step process, and mentions using NLTK.

1. N-gram modeling
2. Traditional text data pre-processing (lower casing, stopword removal, stemming and/or lemmatization, punctuation removal)
3. Feature extraction, specifically mentioning TF-IDF
4. Classification via 6 different models:
    a. Stochastic Gradient Descent
    b. SVM
    c. Linear SVM
    d. KNN
    e. Decision Trees
    f. Logistic Regression

The authors used an 80-20 split to create training and testing sets, and utilized 5-fold cross validation to train a model that predicted whether an article was fake news or real news. The conditions in this study included:

- N-gram variations from unigrams through quadgrams.
- TF and TF-IDF for feature extraction
- Number of features ranging from 1,000 to 50,000

Their findings showed that model performance (as measured only by accuracy) changed with the size of the N-gram, the number of features, and the feature extraction method. Generally speaking, for unigrams, linear methods outperformed non-linear methods across most feature set sizes when using term frequency feature extraction. Otherwise, as the size of the N-gram increased and as the number of features increased, non-linear methods performed equally to or slightly better than linear methods regardless of feature extraction method.

Recommendations

The following are not exhaustive, and based only on my reaction to these articles and personal interest.

1. Use a broader stylometric approach, including N-grams as well as various text analytics, (Example: lengths and average words and characters per sentence).

    a. Some of these metrics have already been calculated as part of the EDA for this data set.
    b. N-grams are important to maintain as they provide context for key words.
2. Include readability and complexity scores.
    a. This will involve choosing the best scoring method, but packages like textstat can help with this.
    b. Many scores do take into consider word length and sentence length, so chosen scores should be based on other metrics to avoid accounting for the same feature twice.
3. Consider using document embeddings, like Doc2Vec, instead of Word2Vec or TF-IDF.
    a. Some research studies have shown a benefit of using Doc2Vec over Word2Vec.
    b. It could also complement additional feature development, like topic modeling when done using Latent Dirichlet Allocation (LDA).

Another potential consideration is to perform topic modeling via pyLDAvis, then use the topic clusters as part of the prediction model. As an unsupervised method, this would be an easily performed task. Similarly, a clustering method based on word embeddings (like K-Means) could also produce another feature for use in the analysis.

Recommended changes to analysis include:
- Creating training, validation, and test sets.
- Test fewer models, but the models selected may change based on the final metrics.
- Expand the metrics used to judge model performance.
    - Keep accuracy, but include specificity and sensitivity, too.
    - METEOR, BLEU, and ROUGE do not seem applicable, and are not recommended at this time.

Toward Hypothesis Development

Recent research combining stylometry and emotion word assessment has been applied to the detection of fake news, and results have proven intriguing.
- Focusing only on stylometry, determine its value as a predictor of fake news vs. real news.

Considering psychology research on topics like framing, cognitive load, and argument pathways, readability and complexity scores may be lower among fake news to make it easier for people to accept the arguments within them.

Some researchers have seen a benefit to using Doc2Vec over its more granular sibling, Word2Vec.

Some topics, like politics or health, might be more likely to have fake news associated with them than others.

Using HuggingFace or OpenAI

- See if I can pass in the same data set
- Cohere and OpenAI have exposed APIs
- See how that performs in terms of fact checking
- See where they are making mistakes, and then compare how the model performs compared to my feature-based prediction model.
- Want to see where they're failing.
- Go for low code frameworks for now.
- Watch Andrew Ng's video
- Try few shot classification.
- Compare and see how the LLMs perform.
- Prompt engineering resources.