Alternus Vera (draft)

Yuhua He
Dept. Software Engineering
San Jose State University
San Jose, CA, USA
Contribution: Corpus Structure
yuhua.he@sjsu.edu

Samuel Yang
Dept. Software Engineering
San Jose State University
San Jose, CA, USA
Contribution: LDA
line 5: email address or ORCID

Yuanzhe Li
Dept. Software Engineering
San Jose State University
San Jose, CA, USA
Contribution: Stance
line 5: email address or ORCID

***Abstract*—Online news, social media and news papers are providing insights for people on make decisions and adjustment. In this paper, we mainly focus on building a machine learning model to make prediction on fake news based on three aspects, corpus structure, LDA and stance of the articles and news content. Separate models are made with proper data cleaning, using NLP distillation process and finally combine features together with a polynomial equation as a final model for giving more accurate prediction as a given news.**

Keywords—ngrams, LDA, stop words, lemmentize, stemming, classification, random forest, logistic regression, pipline, TF-IDF

I. Introduction

Fake news detection has been brought up a significant attention into the public and large number researches has been conducted with online news, social media feeds, news blogs. In this paper, we developed a polynomial equation made of vectors containing three aspects by analyzing corpus structure of content, topic models by applying LDA and stance analysis of the article and news content.

II. Dataset

A. Liar-liar

Liar liar pants on Fire Dataset has 3 files, test, training and valid, each contains 14 columns: ID of the statement, label, statement, subjects, speaker, speaker's job title, state info, party affiliation, total credit history count and context

B. News articles

We selected serval reputable publications for our real news articles dataset as well as several other 'less-reputable' publications in order to have a more balanced representation of news published by a real news publication company.

C. Fake news

Fake news dataset from Kaggle containing uuid, ord_in_thread, author of story, date published, title of the story, text of story, data source, etc

III. Data cleaning

Data cleaning and text preprocessing of the raw contents, containing the following steps

A. Remove Special Characters and Punctuations

B. Lower case the news

C. Tokenization

D. Remove stop words

E. Lemmatization

F. Stemming

G. Spell Check

IV. Feature model construction

A. Corpus Structure

We mainly analyzing the corpus structure based on ngrams

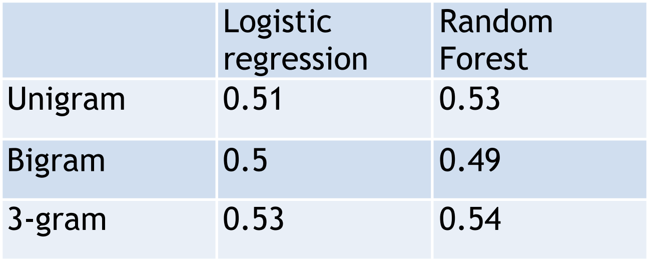
1) Separated the corpus as unigram, bigrams derived from bag of words presentations of each news article content
2) Vectorizing the documents with TF-IDF
3) Applying classification algorithms, logistic regression, random forest for each set of content data
4) Build pipelines to output the accuracy of the model
5) Enriching the original liar-liar dataset with real news and fake news then feed into the pipeline and compare the model performance

- ***What Did I try***

Applying unigram, bigrams, 3-grams, etc from cleaned content, vectorizing the documents with TF-IDF, then apply multiple classification algorithms such as Logistic Regression, Random Forest, etc. Comparing the F1 scores for each. Vectorizing each document based on ngrams do give me the the improvements on the accuracy model provides, also played around with the C values passed in LogisticRegression class, which also gives me slight improvements on the accuracy score.

- ***What Worked***

Analyzing article contents with ngram and applying different classification algorithms do give difference on the model performance, range from 50% - 60% accuracy



	Logistic regression	Random Forest
Unigram	0.51	0.53
Bigram	0.5	0.49
3-gram	0.53	0.54

- ***What Did not work***

Even though ngrams gave the difference on the model accuracy, but with original liar-liar dataset, the model

Identify applicable funding agency here. If none, delete this text box.

accuracy didn't have too much difference, it ranges between 49% - 54%, even though I applied different classification algorithms, the model accuracy still not improved that much.

- ***What alternatives did you try***

I tried data enrichment by getting additional real news data and fake news data from kaggle, cleaned the feature content and text columns by applying and similar cleanning logic used in the original data set, enriched with the orignal dataset and feed into the same model again, and got significant improvement on the model accuracy, it goes up to 87%(logistic regression), 89%(random forest) with unigram.

- ***What did you research, what references (eg code) did you study or leverage (code)***

Automatic Detection of Fake News[1] provides insights in great details on analyzing linguistic features including Ngrams, Punctuation, psycholinguistic features and readability. A Retrospective Analysis of the Fake News Challenge Stance Section Task[2] indicates more on the feature analyzing with RNN model.

- ***What steps did you take as a team to find data enrichment sources, label data, feature engineering***

I toke fake news data[3] from Kaggle as enriched dataset, applying with NLP life cycle: stemming, lemmatization, spelling check, and feed the cleaned data back into the pipeline for vectorizing and build classification model using both logistic regression and random forest.

1. Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea, Automatic Detection of Fake News *(references)*
2. Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, Iryna Gurevych, A Retrospective Analysis of the Fake News Challenge Stance Section Task *(references)*
3. *https://www.kaggle.com/mrisdal/fake-news*