Fake News Detection: Scope Document

Team 18

Under the guidance of Prof. Vasudeva Varma and Mr. Vijayasaradhi I.

Atreyee Ghosal
20161167
Computational Linguistics Dual Degree

Shubhangi Dutta
2018113004
Computational Natural Sciences Dual Degree

Soumalya Bhanja
2019201004
M.Tech in Computer Science and Engineering

Sagnik Gupta
2019201003
M.Tech in Computer Science and Engineering

September 27, 2020

Abstract

With the exponential growth of data and information on the internet, detection of false or fake news is of utmost importance. Fake news mainly consists of untrue or deceptive information, presented as news. It may be directed towards damaging the reputation of an organisation, group or important individuals. It may also be used as a source of money via advertising revenue. This project aims to construct and test an automated detection mechanism of fake or deceptive news.

1 Problem Statement

In this project, we propose to study the fake news detection problem in news articles. Based on the various existing approaches, we formulate it as a text classification or a natural language inference problem. We intend to experiment with different linguistic approaches using multiple word vectorization modules. The source credibility can also be used as one of the major classification criteria for detecting fake news. We compare the language of real news with that of satire, hoaxes, and propaganda to find linguistic characteristics of untrustworthy text.

2 Dataset

We perform experiments using 2 datasets :

- Several27 dataset : It has about 9,408,908 articles and is divided into 10 main groups or tags :
  - Fake news
  - Satire
  - Extreme Bias
  - Conspiracy Theory
  - Junk Science
  - Hate News
  - Clickbait
  - Proceed With Caution
  - Political
  - Credible
- NELA-GT-2019 : It has about 1.2M articles and is divided into 4 main groups or labels:
  - Reliable
  - Unreliable
  - Mixed
  - Unlabelled

3 Evaluation Metrics

In our approach, we treat the fake news detection as a classification problem. Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. To evaluate the performance of our model, we mainly use Accuracy metrics along with other metrics like F1 score, Precision and Recall, against the scores of the baseline.

We use the confusion matrix, to get a more insightful picture of the performance of our model. It helps us in judging which classes are being predicted correctly and incorrectly, and what type of errors are being made.

4 Relevant Work

Various detection techniques have been introduced by the authors who have done significant works in this field. The following are some of the successful techniques found in these works.

- Counts of specific words/specific types of words
  - Verb count
  - Noun count
  - Punctuation count
  - Counts of specific words like "observed" vs. "felt"
  - POS tagging and the counts of each type of tag
  - NE types and the counts of each type
- Shallow syntax and deep syntax
- Combining unigram/bigrams with syntax
- Rhetorical structure framework/analysis
- SVMs/Naive Bayes as baselines
- Sentiment analysis
- Specific feature additions (such as linguistic features) to doc vectors in LSTMS
- Text-only classification vs Text + metadata classification
- Semi-supervised methods: train classifier on a small amount of supervised data, then human-select data that improves performance. Therefore the architecture itself includes broad filtering and downstream filtering
- Regression with checking against ground truth facts
- Generate/Extract central claim + verify central claim (Claim Classification) (Along with ranking to generate a claim)
- Attention/dual attention models

5 Baseline

We use a combination of the following layers as a baseline to evaluate our final model against.

- Word Embedding Layer: We use word2vec to generate word embedding as it is a common model for the same.
- Naive Bayes layer: We use a Naive-Bayes based classifier as it is commonly taken as the baseline [3], [4].

6 Architecture Proposed: Improvements upon the Baseline

The baseline is a naive-Bayes architecture. We seek to improve on it by the following approaches:

- RNN layer: we use an n-layer LSTM as an encoder to form the document representation.

- Classifier layer: Use a multiclass classifier layer to classify the document as one of 4 categories- using the NELA-GT-2019 categories. We improve on the base LSTM by:
  - Using attention-based mechanisms.
  - Using the output of the encoder LSTM as input to the classifier, but with the following additional features appended to the vector:
    * A speaker credibility score for the article
    * Unigram counts of all the words as well as type of words in the document
- We will also attempt to treat this as a claim extraction and verification problem. We extract the central claim of the document using extractive summarization methods, and verify the claim- classifying it as 'true' or 'false' - to give us an idea of the unreliability of the article as a whole.

References

[1] Automatic Deception Detection: Methods for Finding Fake News, *Nadia K. Conroy, Victoria L. Rubin, and Yimin Chen,* 2015 (https://asistdl.onlinelibrary.wiley.com/doi/epdf/10.1002/pra2.2015.145052010082).

[2] A Survey on Natural Language Processing for Fake News Detection , *Ray Oshikawa, Jing Qian, William Yang Wang,* LREC 2020 (https://arxiv.org/abs/1811.00770).

[3] And That's A Fact: Distinguishing Factual and Emotional Argumentation in Online Dialogue, *Shereen Oraby, Lena Reed, Ryan Compton,Ellen Riloff†, Marilyn Walker and Steve Whittaker*, (https://www.aclweb.org/anthology/W15-0515.pdf)

[4] The Lie Detector: Explorations in the Automatic Recognition of Deceptive Language, *Mihalcea & Strapparava,* 2009, (https://www.aclweb.org/anthology/P09-2078.pdf)