Fake News Detection with Semantic Features and Text Mining

Pranav Bharadwaj, Zongru Shao

Fake News Detection with Semantic Features and Text Mining

Pranav Bharadwaj
South Brunswick High School
pranb01@gmail.com

Zongru Shao
Spectronn
zdshao@spectronn.com

Abstract

Nearly 70% of people are concerned about the propagation of fake news. This paper aims to detect fake news in online articles through the use of semantic features and various machine learning techniques. In this research, we investigated recurrent neural networks vs. the naive bayes classifier and random forest classifiers using five groups of linguistic features. Evaluated with *real_or_fake* dataset from kaggle.com, the best performing model achieved an accuracy of 95.66% using bigram features with the random forest classifier. The fact that bigrams outperform unigrams, trigrams, and quadgrams show that word pairs as opposed to single words or phrases best indicate the authenticity of news.

1 Introduction

Nearly 70% of the population is concerned about malicious use of fake news (CNBC)[1]. Fake news detection is a problem that has been taken on by large social-networking companies such as Facebook and Twitter to inhibit the propagation of misinformation across their online platforms. Some fake news articles have targeted major political events such as the 2016 US Presidential Election and Brexit (Kowalewski, 2017). These news articles often rise in popularity and affect the opinions of many readers. Political scientists discovered that between October 7 and November 14, 2016, 27% of adults in the census visited an identified pro-Trump or pro-Clinton fake news website and those fake news consisted approximately 2.6% of the news that they consumed over this period (Sarlin, 2018). External influencers are also creating large masses of automated accounts, called bots, that spread misinformation. It is discovered that approximately 126 million American Facebook accounts have seen fake news from a Russian source sometime in a two year span from 2015 to 2017 (Sabur, 2017). The rising fake-news problem has become increasingly important because of the vulnerability of the massive readers and its wide-spread malicious influence. Consequently, automated identification of fake news has been studied by Facebook and Twitter (Romano, 2018) as well as other researchers.

Several solutions were proposed for this problem. In prior studies, (Tacchini et al., 2017) employed logistic regression and "boolean crowd-sourcing algorithms" to detect fake news in social networking websites. However, this research assumed that agents who post misinformation can be detected by the users who have prior contact of the content (Tacchini et al., 2017). (Wang, 2017) used convolutional neural networks (CNNs), with a long short term memory (LSTM) layer to detect fake news by the text context and additional metadata. Shu et al. (2017) studied linguistic features such as word count, frequency of words, character count, and similarity and clarity score for videos and images while proposing rumor classification, truth discovery, click-bait detection, and spammer and bot detection. Rubin et al. (2015) proposed to classify fake news as one of three types: (a) serious fabrications, (b) large-scale hoaxes, (c) humorous fakes. It also discussed the challenges that each variant of fake news presents to its detection (Rubin et al., 2015).

However, none of the prior studies had utilized recurrent nerual networks (RNNs), naive bayes, or random forest. In this paper, we present a comparison between recurrent neural networks, naive

[1] https://www.cnbc.com/2018/01/22/nearly-70-percent-of-people-are-worried-about-fake-news-as-a-weapon-survey-says.html

bayes, and random forest algorithms using various linguistic features for fake-news-detection. We use the *real_or_fake* dataset from `kaggle.com` evaluate these models.

The remainder of this paper is structured into three sections. Section 2 describes the semantic features and machine learning algorithms in our experiment. Section 3 presents the evaluation results, in which random forest with bigram features achieved the best accuracy of 95.66%. Section 4 presents the conclusions and future work.

2 Materials and Methods

In this section, we describe the material dataset, text preprocessing, semantic features including term frequency (TF), term frequency–inverse document frequency (TFIDF), bigrams, trigram, quadgram, vectorized word representations, and machine learning algorithms such as naive bayes classifier, random forest, recurrent neural networks (RNN).

2.1 Dataset

We use *kaggle real-or-fake news* dataset [2] in our experiments to evaluate semantic features. It contains 6256 articles including their titles. 50% of the articles are labeled as *FAKE* and the remaining as *REAL*. Therefore, detecting the *FAKE* articles is a binary classification problem. We split the dataset into 80% for training and 20% for testing.

2.2 Text Pre-processing

We pre-process the raw text to extract semantic features for machine learning. We use n-grams as semantic features. We first tokenize the title and body of each article. Then, each token is transformed into lower cases and proper nouns lose there capital-letter information. Next, we remove stopwords and numbers for unigrams since they carry less meaning in the context. As a result, the remaining tokens are semantic representations from linguistic perspective. Stopwords and numbers are reserved for n-grams other than unigrams. Then, we extract TF and TFIDF numerical features using the semantic representations.

2.3 Linguistic Features

2.3.1 TF and TFIDF

Note that a text subject (e.g., an article) is called a document in natural language processing. TF computes how frequently a term appears in a document. Given a document d with a set of terms $T = \{t_1, t_2, ..., t_M\}$, and the document length is N (the total occurrence of all terms); suppose term t_i appeared x_i times; then, TF of t_i is denoted as

$$TF(t_i, d) = \frac{x_i}{N} \tag{1}$$

As a result, $[TF(1), TF(2), ..., TF(i), ..., TF(M)]$, $i \in [1, M]$ is a semantic representation for the document. Inverse term frequency (IDF) denotes the popularity of a term across documents. Given a set of documents $D = \{d_1, d_2, ..., d_K\}$ as the subjects of interest, and $TF(i)$ for term t_i is calculated for each document; suppose $C(i)$ denotes the number of documents in which $x_i \neq 0$; then,

$$IDF(t_i, D) = \frac{K}{C_i} \tag{2}$$

Note that each term appears at least once in D. Meanwhile, TF and IDF are calculated in logarithmically scaled:

[2] https://www.kaggle.com/rchitic17/real-or-fake

$$TF(t_i, d_j) = \log \frac{x_i}{N}$$
$$IDF(t_i, D) = \log \frac{K}{C_i}$$

Where $i \in [1, M]$ and $j \in [1, K]$. Then, TFIDF is the product of TF and IDF:

$$TFIDF(t_i, d_j) = TF(t_i, d_j) \times IDF(t_i, D) \tag{3}$$

2.3.2 N-grams

N-grams are continuous chucks of n items from a tokenized sequence for a document. Especially, unigrams are terms where $n = 1$. Bigrams are pairs of adjacent terms where $n = 2$. Trigrams and quadgrams are three and four continuous terms, respectively. For example, the sentence "Your destination is 3 miles away" is tokenized into "your", "destination", "is", "3", "miles", and "away", where each term is a unigram. The bigrams are two-term strings: "your destination", "destination is", "is 3", "3 miles", and "miles away". Trigrams are three-term strings: "your destination is", "destination is 3", "is 3 miles", and "3 miles away". And quadgrams are four-term strings: "your destination is 3", "destination is 3 miles", and "is 3 miles away". In our experiment, we use unigrams, bigrams, trigrams, and quadgrams to calculate the correlated TF and TFIDF features.

2.3.3 Naive Bayes Classifier

The naive Bayes classifier is a classifier based on Bayes' Theorem:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{4}$$

Where A and B are two conditions. The naive Bayes classifier takes each semantic feature as a condition and classify the samples with the highest occurring probability. Note that it assumes that the semantic features are independent (Patel, 2017).

2.3.4 Random Forest Classifier

A decision tree is a "tree" where different conditions branch off from their parents and each node represents a class for classification. The random forest classifier is an ensemble method that operates a multitude of decision trees and thus improves the accuracy. We adjust parameters such as *max_depth*, *min_samples_split*, *n_estimators*, and *random_state* to achieve the best performance; where *Max_depth* is the maximum depth of a decision tree; *Min_samples_split* is the minimum amount of samples to split an internal node, and *N_estimators* is the number of decision trees in the random forest (Breiman, 2001).

2.4 GloVe

GloVe is an unsupervised learning algorithm that parallels the closeness of two words with their distance in a vector space (Pennington et al., 2014). The generated vector representations are called word embeddings. We use word embeddings as semantic features in addition to n-grams because they represent the semantic distances between the words in the context.

2.4.1 RNN

Recurrent neural networks (RNNs) utilize "memory" to process inputs and are widely used in text generation and natural language processing (Mikolov et al., 2010). Long short-term memory (LSTM) is a RNN architecture that uses "gates" to "forget" the input at a condition. In our model we use 100 LSTM

cells in one layer and a softmax activation function. We train the model with 22 epochs and a batch size of 64.

3 Experimental Results

This section presents the experimental results using naive bayes, random forest, and RNN with six groups of features: TF, TFIDF, frequency of bigrams, trigrams, and quadgrams, and GloVe word embeddings.

	TF	TFIDF	Bigram	Trigram	Quadgram	GloVe
Naive Bayes	88.08%	89.90%	90.77%	90.06%	89.74%	-
Random Forest	89.03%	89.34%	95.66%	94.71%	89.60%	-
RNN	-	-	-	-	-	92.70%

Table 1: Accuracy Evaluation

Table 1 shows the accuracy using each method. Observe that random forest results in better accuracy than the naive bayes classifier with TF, bigrams, trigrams, and quadgrams. Meanwhile, bigrams outperform TF, TFIDF, trigrams, and quadgrams. The RNN model with GloVe features outperform TF, TFIDF, and quadgrams but not bigrams and trigrams.

Note that unigrams represent words; bigrams represent words and their one-to-one connections; trigrams carry level-two connections for words if consider a one-to-one connection between two words as level-one. As a result, bigrams carry more information than unigrams; trigrams more than bigrams; and quadgrams more than trigrams. Also, more information for training suppose to provide better accuracy. Therefore, we would expect quadgrams to result in higher accuracy than trigrams, trigrams higher than bigrams, and bigrams higher than unigrams. This assumption contradicts with Table 1 given that quadgrams do not result in the highest accuracy. The reason is when information increases, the training process takes specific details and the model is "over-fitted". However, bigrams do outperform unigrams because they carry more information. For the same reason, TF and TFIDF result in similar accuracies because they both are derived from unigrams. Meanwhile, GloVe with RNN outperforms unigrams and quadgrams but results in lower accuracy than bigrams and trigrams. This is caused by single layer LSTM cells and word embeddings represent unigrams. Therefore, the RNN model outperforms random forest if disregard the difference between word embeddings and unigrams.

4 Conclusion

In this paper, we applied semantic features including unigram TF & TFIDF, bigrams, trigrams, quadgrams, and GloVe word embeddings along with naive Bayes, random forest, and RNN classifiers to detect fake news. The performance is promising as bigrams and random forest achieved an accuracy of 95.66%. It implies that semantic features are useful for fake news detection. As the next step, semantic features may be combined with other linguistic cues and meta data to improve the detection performance.

References

Breiman, L. (2001). Random forests machine learning. 45: 5–32. *View Article PubMed/NCBI Google Scholar*.

Kowalewski, J. (2017). The impact of fake news: Politics. *Lexology*.

Mikolov, T., M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur (2010). Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Patel, S. (2017). Chapter 1 : Supervised learning and naive bayes classification - part 1 (theory). *Medium*.

Pennington, J., R. Socher, and C. Manning (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Romano, A. (2018). Mark zuckerberg lays out facebooks 3-pronged approach to fake news.

Rubin, V. L., Y. Chen, and N. J. Conroy (2015). Deception detection for news: three types of fakes. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, pp. 83. American Society for Information Science.

Sabur, R. (2017). Facebook to tell users how much russian fake news they have been exposed to.

Sarlin, B. (2018). 'fake news' went viral in 2016. this expert studied who clicked.

Shu, K., A. Sliva, S. Wang, J. Tang, and H. Liu (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter 19*(1), 22–36.

Tacchini, E., G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro (2017). Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*.

Wang, W. Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.