Fake New Challenge (FNC-1): Stance Detection

COMP9417 Machine Learning Project

Shashank Reddy (z5222766), Leonard Lee (z5173917), Connor Mcleod (z5058240), Darren Zhang (z5113901)

Files for this submission can be downloaded here

1. Introduction

Fake news is a recent trend which describes the spreading of disingenuous or deliberately falsified information to consumers. This false information can be distributed in many different forms but is usually found in printed and social media (Webwide, 2019). Several factors have contributed to the rise of fake news in recent years, one of the most controversial being the Russian interference in the 2016 U.S. election. The spread of misinformation and propaganda can have a significant impact on their subject matter, and so it is vitally important that fake news is identified so that it's reach can be limited.

The 'Fake News Challenge', FNC-1, is an original attempt to develop fake news identification software using machine learning principles. The goal of the challenge was to 'explore how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem' (Fake News Challenge). The challenge was conducted in 2016/17 and several teams from around the world took part.

The challenge focused on detecting whether the body of an article matched the context of the headline. By examining the text contained in both, it is possible to classify the article's body as being 'unrelated', 'discussing', 'agreeing' or 'disagreeing' with the article's headline.

Our objective is to complete FNC-1 using several learning methods to compare the results of each - namely the Naive Bayes (NB), logistic regression (LR), Decision Tree Classifier (DT) and Random Forest Classifier (RF) models - using the provided datasets. The developed models will be tested on the competition dataset to assess their prediction accuracy.

2. Related Work

This section will explore the implementations and research of others into the fake news problem. We have incorporated the text-classification concepts developed by others into our solution in order to learn from our peers and create a solution that incorporates the latest text-classification principles.

2.1 FNC-1 results

Over 50 teams participated in FNC-1, with the top four teams scoring accuracy results of over 80%, beating the competition provided baseline result of 75.20%. The top results are shown in Table 1 below:

Team	**% FNC-1 score**
SOLAT in the SWEN	82.02
Athene	81.97
UCL Machine Reading	81.72
Chips Ahoy !	80.21
FNC-1 baseline	**75.20**

Table 1: FNC1 Competition results

The winning team, representing Cisco systems, won the competition using a 50-50 weighted average combination of convolutional neural networks (CNN) and gradient-boosted decision trees (GBDT). The CNN used two one-dimensional convolutional neural networks which fed into a multi-layer perceptron (MLP) with 3 hidden layers. For the GBDT, they used 5 overarching sets of features passed into the decision trees (Largent, 2019).

2.2 Literature review

There is a growing field of research into text-classification problems, particularly related to fake news identification. One important facet of text-classification problems is understanding how best to pre-process data for the inputs to the learning models. Stop word removal, the removal of extraneous words, and stemming, reducing words to their morphological root, have been recommended as useful techniques prior to modelling (Ahmed, et al.).

Several learning models have also been investigated in research. Linear-based classifiers have been shown to provide good results for text-classification problems (Ahmed, et al.). Tacchini, Ballarin, Vedova, Moret, & Alfaro used logistic regression and harmonic boolean label crowdsourcing to confirm their hypothesis that user interaction with future posts can predict whether those particular posts are hoaxes (Tacchini, et al.). They effectively concluded that BLC algorithms had a 99% accuracy when trained with at least 80 posts and showed that these algorithms could be applicable for real-world use.

Masood, Razan & Aker documented how problem-specific feature engineering and deep learning models could be used to produce solutions to the fake news problem (Masood, et al.). They implemented multi-step classifier settings using traditional machine learning approaches. Combining different learning algorithms to the problem meant that their results outperformed the competition winner's model.

3. Implementation

This section will describe our solution to the FNC-1 problem. It explains the data preprocessing, feature extraction and learning models used.

3.1 Pre-processing

Preprocessing data is the first step when working with a new dataset. It is important to clean the data by removing and redundancies to ensure that the extracted information is useful inputs to the models. This also reduces the size of the data processed, meaning that classifying it will be more efficient, and ideally, accurate. We employed several preprocessing techniques on the FNC-1 dataset, each of which is explained below.

Tokenizing
In order to carry out further processing, the long text has to first be broken down to singular words. This process is called word tokenizing. We used the Natural Language Tool-Kit (NLTK) with Python to tokenize the dataset.

Removal of stop words, alpha-numerics and punctuations
Stop words are extra words in the english language that are not important for the text classification process. Examples of common stop words are: *a, an, are, at, by, for, in, is, of, on, that, the, this, too, what, where, who*. In addition to removing stop words, alphanumerics and punctuation are removed, and all of the text is converted to lower-case.

Stemming and Lemmatization
Stemming and lemmatization helps to transform the tokens into its standard or root form and decreases the number of words in the data. An example of this is reducing all of ‘running’, ‘ran’, ‘runner’ to ‘run’. This reduces the number of features input to the models, reducing the time-cost of the algorithm.

3.2 Feature Extraction

After preprocessing, the text has been refined down to a list of words that provide useful summary information about the entire text. However, there is usually still a large number of words, meaning running the models on these will have a high computational burden. Further reducing the features using a process called feature extraction can help to reduce the large feature space dimensions and increase the accuracy of the classifiers. The feature extraction methods used in our solution are described below.

Term Frequency - Inverse Document Frequency (TF-IDF)
TF-IDF is a weighting metric used to measure the importance of a word in a document. It’s value increases with the number of times a word appears in that document and will be counteracted by the frequency of the word in the corpus, or entire set of documents.

$$\mathrm{tf}(t,d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)$$

N-gram
Another popular feature identification approach used in natural language processing is using N-gram modeling. N-gram is a contiguous sequence of items with length n. In this case, word-based n-gram is used to represent the context of the document and generate features. Texts are changed to lists of unigrams, bigrams and trigrams which are various groups of n-number tokens.

Sentence Weighting
In sentence weighting, the body of the text is compared to the headline and the common words counted between them, known as the intersection. The union is the total number of words in both the body and headline. The weight of the sentence is calculated by dividing the intersection by the union. The result will be in the range of 0 to 1, where the higher the result, the more the headline relates to the body with similar texts.

3.3 Training models

Several training models were developed in order to compare the results between different classification algorithms.

3.3.1 Naive Bayes

Bayes' theorem suggests that the probability of an event occurring is based on the prior probability of conditions related to that event, and is given by the following equation:

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

In the case of the FNC-1 challenge, A is the stance of the article, and B is one word of the article.

We have applied the Naive Bayes (NB) classifier to the FNC-1 challenge to see how a simple learning algorithm can hold up against more complex algorithms. The key assumption of the NB classifier is that all features or inputs (article words) are independent of each other. This is not true for coherent news articles, where the words are not independent events. Despite this, NB classification has typically performed well in text classification problems and sets a good baseline result for our comparisons.

3.3.2 Logistic regression

Logistic regression (LR) is a supervised learning algorithm that develops a model that predicts the values of a target from a discrete set (Ziegel and Menard, 1996). Similar in function to linear regression initially, this model then applies the 'sigmoid' function which maps an output to between 0 and 1. A decision boundary is then determined to classify different values as a discrete result.

3.3.3 Decision tree Classifier

Decision trees (DT) are a commonly used supervised learning algorithm for regression and classification problems. Decision trees represent a decision framework that works down the tree's nodes - or features - making decisions based on these nodes to come to a final classification result. The decisions are learned from the supervised training set and applied to the test set. Decision trees can have difficulty with text classification if there is a large amount of equally weighted feature inputs.

3.3.4 Random tree Classifier

Random Forest, like Decision Tree Classifier can be used for solving both regression and classification problems. It is an ensemble learning model. (Breiman, 2001) The algorithm creates forest with several Decision Trees. More the trees in the forest, higher the accuracy of the classifier.

In this algorithm, **k** features are randomly selected from a total of **n** features.

- Among the **k** features, using the best split point node **d** is calculated.
- The node is now split into daughter nodes using the best split.
- The above steps are iterated until **l** number of nodes are formed.

Repeat the above 4 steps m times until m decision trees are formed. Two important features of RF are the ability to achieve high prediction accuracy and usability of desired capabilities.

4. Results

We ran the four models on the FNC-1 dataset to identify which worked best. The results found that the random-forest model performed the best when tested on the competition dataset, as shown in Table 2. The accuracy is a measure of how many predictions are correct out of the total number of articles. The competition score is the custom scoring provided by the competition, where the percentage is the model's score divided by the maximum possible score. The validation score is the accuracy of the model when applied to the validate subset of the training dataset. Further performance results for each model can be found in Appendix A.

Model	Naive Bayes	Logistic regression	Decision tree	Random forest
Competition accuracy	69.60%	80.90%	83.40%	84.40%
Competition score	45.12%	72.40%	73.07%	74.39%
Validation accuracy	62.70%	80.10%	79.50%	85.4%
Validation score	54.50%	78.13%	71.88%	80.36%

Table 2: FNC-1 model results

Naive Bayes performed well on the training and validation sets, however, did not perform as well on the competition test set. This indicates that the NB implementation was probably overfitting the training data. The main weakness of the NB model was incorrectly classifying most articles as 'unrelated', meaning a high number of false positives.

The LR, RF and DT models all performed well, with accuracies of over 80% on the competition dataset. Each of these models had difficulty classifying an article as disagreeing with the headline.

Figure 2 below summarizes the stance predictions of each model vs the actual stance for the competition dataset.

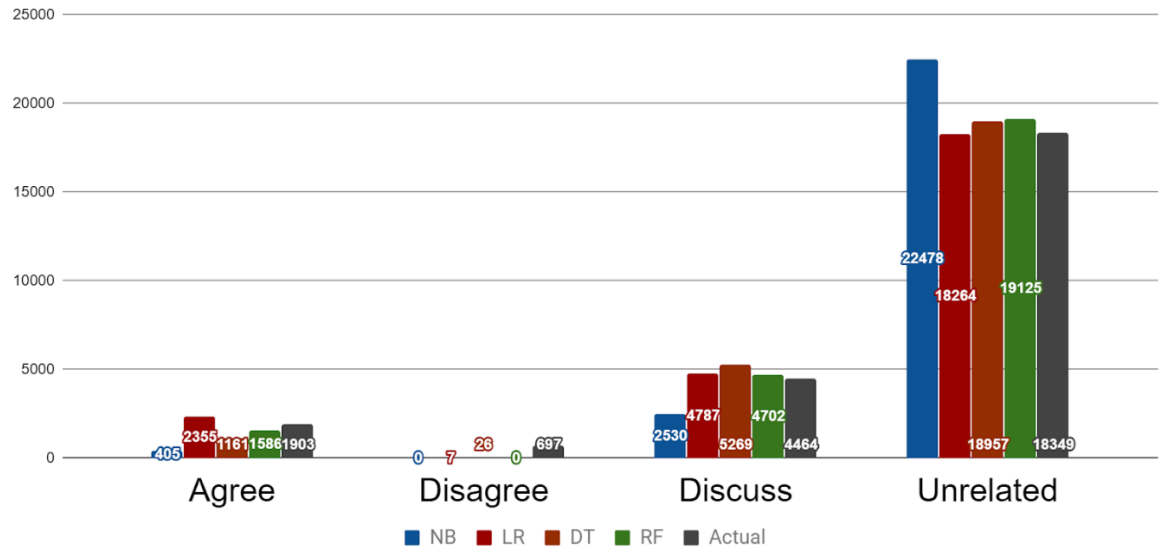

Figure 2: FNC-1 model predictions vs actual

5. Future Work

Our results have shown how different learning algorithms perform in a text classification problem. The trained models perform moderately well when applied to the competition dataset, however, there is room for improvement for them to be considered functionally useful. The purpose of our research was to compare the results between different learning algorithms and to further our knowledge of their practical implementation. Future work might involve optimizing the models we've used for higher accuracy or faster prediction speeds. A more advanced approach may be to combine models, improve data pre-processing for universal inputs, or implement other novel approaches to make the results commercially useful.

6. Conclusion

We have implemented a unique solution to the Fake News Challenge. We did this by incorporating a range of different preprocessing and feature extraction techniques and applying several commonly used supervised learning algorithms to the processed data. We found the relative performance of the five models, understanding better their strengths and limitations, and this allowed us to pinpoint and identify the most effective algorithms for detecting fake news. After analyzing and understanding the results, it appears that Random-forest algorithms show promise for use in real-life applications for detecting fake news, but one study alone is not enough to decisively measure the efficiency of one algorithm.

Appendix A: Model accuracy results

Three additional performance metrics were determined for each of the models' predictions. They are precision, recall and F1-score - all commonly used performance measures for determining the predictive ability of a classification algorithm. The metrics are defined as the following:

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|X_i \cap Y_i|}{|Y_i|}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|X_i \cap Y_i|}{|X_i|}$$

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|X_i \cap Y_i|}{|X_i| + |Y_i|}$$

Where Xi is the set of predicted labels, Yi is the set of true labels and n is the number of samples.

Precision	Naive Bayes	Logistic regression	Decision tree	Random forest
Agree	0.3257	0.5254	0.4639	0.6060
Disagree	0.2406	0.3206	0.0510	0.9230
Discuss	0.5371	0.7119	0.6401	0.7564
Unrelated	0.7352	0.9269	0.8977	0.9292

Table 3: FNC1 Precision Results

Recall	Naive Bayes	Logistic regression	Decision tree	Random forest
Agree	0.3474	0.6216	0.1782	0.4841
Disagree	0.5066	0.5502	0.0218	0.1048
Discuss	0.4591	0.7467	0.7619	0.8290
Unrelated	0.7458	0.8598	0.9350	0.9490

Table 4: FNC1 Recall Results

F1-score	Naive Bayes	Logistic regression	Decision tree	Random forest
Agree	0.3362	0.5695	0.2575	0.5382
Disagree	0.3263	0.4051	0.0306	0.1882
Discuss	0.4950	0.7289	0.6957	0.7910
Unrelated	0.7404	0.8920	0.9120	0.9390

Table 5: FNC1 F1 Score Results

References

Fake News Challenge, ‘Exploring how artificial intelligence technologies could be leveraged to combat fake news’. From: http://www.fakenewschallenge.org/

Webwise.ie. (2019). Explained: What is Fake news? | Social Media and Filter Bubbles. [online] Available at: https://www.webwise.ie/teachers/what-is-fake-news/

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tom Kocisky, and Phil Blunsom. 9 2015. Reasoning about Entailment with Neural Attention.

Ahmed, Hadeer & Traore, Issa & Saad, Sherif. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. 127-138. 10.1007/978-3-319-69155-8_9.

Tacchini, E., Ballarin, G., Vedova, M. L. D., Moret, S., Alfaro, L. (2017). Some Like it Hoax: Automated Fake News Detection in Social Networks. Retrieved August 9, 2019, from: https://arxiv.org/pdf/1704.07506.pdf

Breiman, L. (2001). Journal search results - Cite This For Me. *Machine Learning*, 45(1), pp.5-32.

Ziegel, E. and Menard, S. (1996). Applied Logistic Regression Analysis. *Technometrics*, 38(2), p.192.

Masood, Razan & Aker, Ahmet. (2018). The Fake News Challenge: Stance Detection Using Traditional Machine Learning Approaches. 10.5220/0006898801280135. From: http://www.insticc.org/Primoris/Resources/PaperPdf.ashx?idPaper=68988

Largent, W. (2019). *Talos Targets Disinformation with Fake News Challenge Victory*. [online] Blog.talosintelligence.com. Available at: https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html