COMP90042 Project 2020:
Climate Change Misinformation Detection

Vihanga Keshawa Jatalath - 1088212

Abstract

An enormous amount of misinformation spreading through sources such as social media has been observed in recent years in the forms of rumours, fake news, opinions mistaken as facts, etc. (Shao, C., Ciampaglia, G. L., Flammini, A., & Menczer, F., April 2016). This can cause major risks in conveying information to the public where reliability is paramount (Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F., 2017). This project aims to classify misinformation regarding climate change in contrast to reliable information of other news using several approaches of Natural Language Processing (NLP). The major caveat in the project is the provided training data contains only misinformation about climate change. Hence several approaches were considered including; a) one class text classification using only the misinformation training data and b) using web scraping to collect reliable data from web sources to perform supervised binary classification. This report discusses how these methods were used and optimised in selecting the most suitable approach for the task.

1 Introduction

In recent years, the speed and ease of use of social media has significantly increased the speed of information conveyed by various parties in social as well as mainstream media. However, the drawback of such ease of use is the reliability of the information conveyed to the public. In the case of *climate change*, the question of whether global warming is human induced remains controversial regardless of the supporting scientific attestation. Li, Y., Zhang, J., & Yu, B. (September 2017) states that even though scientific information exists regarding certain topics, the methods in which this information are communicated vary from source to source arguing the methods in which these sources convey and how the audience interpret such information leads to massive impacts on their behaviour leading to fact-checking of these information to be paramount.

Due to the massive amount of content published online per day, advanced methods of fact checking are required to detect such misinformation in contrast to traditional fact checking methods (Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F., 2017). This study examines several methods to perform this task using a dataset of climate change misinformation to perform binary classification to classify whether a given article is misinformation or not.

The main caveat of this task is that the provided training dataset consisted only of climate change misinformation. One method in dealing with this issue is using single class (one class) classification machine learning model with the climate change misinformation dataset, where a given article is considered not to be misinformation if it is classified as an outlier by the model.

Another approach to the task was creating a dataset which consisted of only reliable information of climate change and use it along with the provided misinformation dataset. This approach paved the way to use more sophisticated machine learning models such as Neural Networks and LSTM (Long-Short Term Memory) (Sherstinsky, A., 2018) in contrast to simple classification models such as support vector machines.

The remainder of this paper discusses about related work on the topic, the datasets used, the learning models in detail, error analysis and evaluation of the approaches used in detail.

2 Background

Several approaches were explored in identifying the best method to classify climate change misinformation including:

One class classification: which used only the provided misinformation dataset. This method is used in binary classification problems where one class is significantly under sampled

compared to the other class (Tax, D. M. J., 2002) which was the case in this project.

Supervised Binary Classification: This approach requires positive and negative training data in order to build a classifier.

According to (Poddar, K., & Umadevi, K. S., 2019), the Support Vector Machine (SVM) classifier performs well in terms of text classification using TF-IDF vectorizer, by comparing performances of using both TF-IDF vectorizer and CountVectorizer in extracting the Bag-of-words (BOW) as the feature set.

Saha, D. (February, 2011) proposes a method for text classification. Using neural networks. In this project, Feed Forward Neural Networks were used in performing text classification using a BOW representation of the training text data as the features of the model.

Another supervised binary classification algorithm used was LSTM (Sherstinsky, A., 2018). Liu, G., & Guo, J. (2019) proposes a bidirectional LSTM which preserves the context of the following and preceding words to a given word compared to traditional LSTMs for text classification in order to improve performance of text classification using Recurrent Neural Networks (RNN).

The remainder of this report discusses how these technologies were used in building the most suitable classifier for this task.

3 Methodology

The approaches in which the discussed algorithms were used and examined are discussed in this section.

3.1 Datasets

Although this dataset was used for the one-class classification approach, in order to build a supervised binary classifier to improve the results obtained via one-class classification approach, the training dataset needed to be expanded to include data not considered to be climate change misinformation in order to create a dataset with negative labels.

Web scraping was used to perform this task via using the *beautifulsoup* and *newspaper3k* web scraping libraries. In using web scraping to create the negative dataset, the reliability of the data sources (in this case news websites) was a major concern. Hence news articles from several reputable sites were collected including sites such as *climate.nasa.gov*, *edition.cnn.com*, *www.abc.net.au*, etc.

In exploring the provided development data set, it was clear that the negative classification data included reliable factual climate change data as well as articles relating to other news topics. Hence in creating the negative training dataset, the majority of climate change data was collected from the *climate.nasa.gov* website where news articles relating to other topics were collected from other reputable news sites such as *cnn.com* and *abc.net.au*.

The total training dataset now consisted of 1168 instances of climate change misinformation (positive) and 900 instances each of climate change related reliable information and other reliable news articles (negative).

3.2 Preparing the datasets

The provided datasets included the positive training data, development data which included 100 instances of both positive and negative data, and test dataset of 1410 instances as JSON files. In order to extract the text data from these files, they were loaded into a python script and converted into 'Data Frames' using the *json* and *pandas* libraries from python.

Once web scraping was performed to generate the negative training dataset, the climate change related and unrelated reliable articles were saved into two more json files. These files were loaded into the python script when performing supervised binary classification, whereas in one-class classification, they were not required.

Once the data was obtained, feature extraction was the next step in solving this task. Since this task relies on the sentence structure and context, very little pre-processing was done in order to preserve these elements. This included selecting only the ascii characters from the texts and removing tags such as '*\n, \t, \r, etc.*'. No forms of lemmatisations, stemming, or lowercasing was performed on the texts since doing so may compromise the sentence structure and contexts of the text data.

One-class Classification: This approach needed only the provided positive dataset since it uses outlier detection to perform classification.

From the processed text data, bag-of-words (BOW) was used as features for the one-class classification algorithm. This was obtained using the *sklearn CountVectorizer* library. Afterwards, these features were fed into *OneClassSVM* model in order to train the data and obtain the predictions.

Supervised Binary Classification: The collected negative dataset, along with the provided positive dataset were used to build and examine multiple supervised binary classification algorithms.

Model	Features
SVM	BOW using word tokens with n-grams using *sklearn CountVectorizer* library
FFNN	BOW using word tokens from *keras* library
LSTM	Word sequences using *keras* library

Table 1 Features used in the models

Table 1 shows the features used for each supervised classification model.

The FFNN model was designed with the vocabulary size of the train dataset as input layer, one hidden layer containing 10 neurons with '*relu*' as the activation function, a dropout layer to reduce overfitting, and a dense layer containing 1 neuron with a sigmoid activation function to obtain the prediction probabilities.

The LSTM model consisted of an input later of the size of the vocabulary, an LSTM later with 10 units, and a dense layer as output with a sigmoid function as the activation function in order to obtain the prediction probabilities of the classifications.

4 Results

4.1 One-class Classification Results

n grams	Metrics			
	Acc	Prec	Recall	F1
(1,1)	0.61	0.59	0.68	0.63
(1,2)	0.66	0.60	0.90	0.72
(2,2)	0.56	0.57	0.46	0.51

Table 2 Evaluation metrics of one-class model dev set predictions

According to Table 2, the best performance of the one-class SVM classification algorithm was obtained using unigram and bigrams when creating the BOW representation.

4.2 Supervised Binary Classification

Using *GridsearchCV* from the *sklearn* library, the best parameters used for the SVM classifier were obtained. Afterwards, several combinations of the BOW features were tested and evaluated based on the development set performance.

n grams	Metrics			
	Acc	Prec	Recall	F1
(1,1)	0.67	0.61	0.90	0.73
(1,2)	0.79	0.73	0.90	0.81
(2,2)	0.83	0.77	0.94	0.84
(1,3)	0.77	0.71	0.90	0.79
(2,3)	0.82	0.75	0.94	0.83
(3,3)	0.77	0.72	0.86	0.78

Table 3 Evaluation metrics of SVM dev set predictions

The FFNN used a similar approach to the SVM classifier in comparing different combinations of n-grams in the BOW feature set to obtain the best performance.

n grams	Metrics			
	Acc	Prec	Recall	F1
(1,1)	0.86	0.81	0.94	0.870
(1,2)	0.79	0.73	0.92	0.814
(2,2)	0.81	0.78	0.86	0.819
(1,3)	0.84	0.84	0.84	0.840
(2,3)	0.84	0.82	0.86	0.843
(3,3)	0.79	0.74	0.88	0.807

Table 4 Evaluation metrics of FFNN dev set predictions

In training the LSTM model with the training dataset, it was observed that the maximum number of word embeddings that could be taken as input to the model to be 500 since exceeding this number resulted in reduces accuracy, precision and f1 score.

Max length	Metrics		
	Prec	Recall	F1
50	0.73	0.76	0.74
100	0.75	0.8	0.77
300	0.84	0.76	0.80
500	0.60	0.86	0.71

Table 5 Evaluation metrics of LSTM dev set predictions

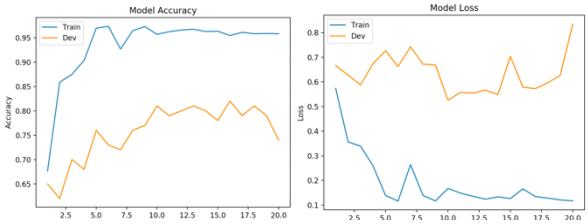

Figure 1 Accuracy and Loss plots over 20 epochs with maxlen = 300

5 Error Analysis

As mentioned in the previous section, the evaluation metrics of the one-class classification model were considered as the baseline metrics for the remaining models to be improved upon. According to the results obtained, using BOW as features, the results of the SVM classifier had a considerable improvement upon the base model. According to table 3, the SVM model was further improved using different combinations of n-grams of the BOW. Although the BOW disregards the sentence structure of words in the document, one could argue that using bigrams mitigates this issue to a certain extent.

The next model tested was the FFNN model. According to table 4, the best evaluation metrics for this model was obtained using unigrams in creating the BOW with the *sklearn CountVerctorizer* library, which gave a better precision value compared to the SVM. Hence when predicting the test dataset from both models, the FFNN resulted in a better f1 score compared to SVM. (Table N)

Model	Metrics		
	Prec	Recall	F1
One Cl	0.24	0.74	0.36
SVM	0.35	0.88	0.50
FFNN	0.47	0.94	0.63

Table 6 Test set evaluation metrics for the models

Index	Label	Pred SVM	Pred NN
dev-14	1	0	1
dev-67	0	1	0
dev-80	0	1	0

Table 7 Dev set predictions for SVM and FFNN

Table 7 shows how incorrectly classified articles by the SVM were correctly classified using the FFNN. One reason for this is the SVM model does not consider word embeddings, whereas the FFNN model automatically generates word embeddings when generating the outputs of each layer. Word embeddings map discrete word symbols into a continuous vector space in a relatively low dimension. This allows the model to capture fine grain relationships of the texts. Hence in this case it was clear that FFNN performed better than SVM.

When comparing the evaluation metrics of the FFNN and LSTM models, according to tables 4 and 5, the FFNN model performed better. This contradicts what most studies conclude in text classification, since LSTM should perform better compared to FFNNs since they take into account the sentence structure and word contexts.

One reason for this maybe due to only 500 word embeddings being considered whereas an average document contained over 1000 words and the maximum length of a document in the provided training set being over 5800 words. In this case, it could be argued that the entire context of the documents was not considered in the LSTM model.

Also, according to Figure 1, it was evident that the LSTM model tended to overfit to the training data since the loss of the dev set increased after a certain number of epochs, resulting in poor performance in the dev set. Several actions were undertaken to avoid overfitting including reducing the complexity of the LSTM via reducing the embedding dimensions, reducing the LSTM units, varying the batch sizes when training, testing with different activation functions such as *sigmoid* and *relu*, etc. However, the performance could not be improved over the FFNN model. These issues could be avoided by using a pre-trained model such as Google's BERT.

6 Conclusion

This project involved performing text classification on a provided dataset in order to identify climate change misinformation from factual information. Among the two main methods used, one-class classification model was chosen as the base model, and several supervised binary classification models were trained and optimised to obtain the best evaluation metrics to select the best performing model.

Future improvements could be made to the system including using pre-trained models such as Google's BERT which uses pre-trained word embeddings and can be used to perform text classification by adjusting the

parameters such that they fit the desired downstream task.

This project can be further improved to design an automated system which retrieves news articles based on a reliability rating that a user defines, creating a news article reliability filtering system.

References

Shao, C., Ciampaglia, G. L., Flammini, A., & Menczer, F. (2016, April). Hoaxy: A platform for tracking online misinformation. In *Proceedings of the 25th international conference companion on world wide web* (pp. 745-750).

Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F. (2017). The spread of misinformation by social bots. *arXiv preprint arXiv:1707.07592*.

Li, Y., Zhang, J., & Yu, B. (2017, September). An NLP analysis of exaggerated claims in science news. In Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism (pp. 106-111).

Khatua, A., Khatua, A., & Cambria, E. (2019). A tale of two epidemics: Contextual Word2Vec for classifying twitter streams during outbreaks. Information Processing & Management, 56(1), 247-257.

Sherstinsky, A. (2018). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. arXiv preprint arXiv:1808.03314.

Tax, D. M. J. (2002). One-class classification: Concept learning in the absence of counter-examples.

Guerbai, Y., Chibani, Y., & Abbas, N. (2012, May). One-class versus bi-class SVM classifier for off-line signature verification. In 2012 International Conference on Multimedia Computing and Systems (pp. 206-210). IEEE.

Poddar, K., & Umadevi, K. S. (2019, March). Comparison of Various Machine Learning Models for Accurate Detection of Fake News. In 2019 Innovations in Power and Advanced Computing Technologies (i-PACT) (Vol. 1, pp. 1-5). IEEE.

Saha, D. (2011, February). Web text classification using a neural network. In 2011 Second International Conference on Emerging Applications of Information Technology (pp. 57-60). IEEE.

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing, 337, 325-338.