COMP90042 Project 2020
Climate Change Misinformation Detection

Name: Zongru Li
Student ID: 947539

Abstract

Aiming at recognising climate misinformation, this system applies to pre-process, information extraction, and multiple machine learning models to solve it. In order to search the better resolution for this problem, four models, which are OneClassSVM, Logistic Regression, FFNN, TextCNN, are used. The performance of these models is discussed and analysed their shortcoming. Finally, several assumptions are put forward for improvement.

1 Introduction

Numerous climate misinformation is released in mainstream and social media, which leads to persistent debate on climate problems. Therefore, distinguishing climate misinformation in an effective way is a way to decrease this kind of debate and make people realise the truth of the climate problems.

The aim of this project is to build a system to detect climate misinformation with machine learning methods. To be specific, pre-process, information extraction, and multiple machine learning models are applied in this project to achieve the goal.

2 Methodology and Implementation

In this section, dataset and approaches applied in this system will be introduced. In this project, the system can be apart into three parts, pre-process, feature-extraction, and machine learning model application. Also, to implement binary classification, external data is introduced in this system. Details are discussed in the following content of this section.

2.1 Dataset

Due to only the positive train data set is provided, two classes classification cannot implement with the original data. Therefore, to implement the binary classifier, the negative train data need to be supplied. About the negative train data, some climate real information(from https://www.metoffice.gov.uk) and news of other aspects, including business, entertainment, politics, sport, and technology, are added in the negative dataset(from https://www.kaggle.com/pariza/bbc-news-summary).

2.2 Pre-process

Before classification, the pre-process is an indispensable process of the whole work. Firstly, all of the words are transfer to lower case and stopwords are removed. Then, the words are lemmatized according to their pos tag and stemmed.

2.3 Feature Extraction

In the part of feature-extraction, TF-IDF model is applied in this system. TF-IDF is short for term frequency-inverse document frequency, which can reflect the importance of a word in the whole text. Focus on the project, the TF-IDF of all words are calculated and transferred to a matrix(Rajaraman & Ullman, 2011).

2.4 Machine Learning Model Application

In order to evaluate which kind of model can get the ideal performance, four models are applied in this project, which are OneClassSVm, Logistic Regression, FeedForwardNN, and TextCNN.

2.4.1 OneClassSVM

For the reason that the original data only has positive samples, one class classification is a choice for solving this problem. OneClassSVM is a kind of one class unsupervised learning algorithm. It creates a non-linear decision boundary by projecting the data through a non-linear function to a space with a higher dimension. In other words, it trains a smallest hypersphere. All of data is all covered in the hypersphere. When identifying a new data, if the data point falls within the hypersphere, it belongs to this category, otherwise it is not(Vlasveld, 2013).

2.4.2 Logistic Regression

After collecting some negative samples, binary classification can be introduced. Logistic Regression is a kind of supervised learning algorithm, which is widely applied to text classification. This learning algorithm uses a logistic function to model a binary dependent variable and calculate the probability of classification for the unlabelled data. Because of its usability and practicability, this algorithm is applied in this project to solve this problem and compare it with neural network algorithms(Tolles & Meurer, 2016).

2.4.3 FeedForwardNN

The feedforward neural network is an artificial neural network, there is no cycle in the connection between nodes. In the feedforward neural network, the information moves from the input layer, through the hidden nodes, and finally to the output nodes, only in one direction(Schmidhuber, 2015).
Specify in this project, three dense layers are set in the model, the first two layers use relu activation function, and the last layer uses sigmoid activation to process classification, with dropout layer and BatchNormalization layer to correct the result.
The specific composition of the model is shown in the figure 1.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 10)	316470
dropout_1 (Dropout)	(None, 10)	0
batch_normalization_1 (Batch	(None, 10)	40
dense_2 (Dense)	(None, 256)	2816
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 2)	514

Figure 1: Composition of FFNN

2.4.4 TextCNN

Convolutional neural networks (CNN) utilize layers with convolving filters that are applied to local features, also, it can be applied in text classification. In this model, the result of each convolution will trigger when a special pattern is detected. By varying kernels' size and concatenating their outputs, the detect patterns of multiples sizes can be changed (2, 3, or 5 adjacent words)(Maheshwari, 2018).
In details, this model consists of 7 types of layer, which is input layer, embedding layer, convolutional layer(1-dimension), max-pooling layer, flatten layer, dropout layer, and dense layer. At first, the dataset needs to be transferred into same length sequence, and then put it into embedding layer, after that, building some convolutional layer with different input channels and maxpooling them. Finally, concatenating these convolutional layers and make flatten and dropout to modify the result. After above mentioned process, using dense layer to classify.
However, although this model is easy and efficient, it has a high computational cost and needs a lot of training data. The specific composition of the model is shown in the figure 2.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 150)	0	
embedding_1 (Embedding)	(None, 150, 150)	4746900	input_1[0][0]
conv1d_1 (Conv1D)	(None, 150, 256)	115456	embedding_1[0][0]
conv1d_2 (Conv1D)	(None, 150, 256)	153856	embedding_1[0][0]
conv1d_3 (Conv1D)	(None, 150, 256)	192256	embedding_1[0][0]
conv1d_4 (Conv1D)	(None, 150, 256)	230656	embedding_1[0][0]
conv1d_5 (Conv1D)	(None, 150, 256)	269056	embedding_1[0][0]
max_pooling1d_1 (MaxPooling1D)	(None, 3, 256)	0	conv1d_1[0][0]
max_pooling1d_2 (MaxPooling1D)	(None, 3, 256)	0	conv1d_2[0][0]
max_pooling1d_3 (MaxPooling1D)	(None, 3, 256)	0	conv1d_3[0][0]
max_pooling1d_4 (MaxPooling1D)	(None, 3, 256)	0	conv1d_4[0][0]
max_pooling1d_5 (MaxPooling1D)	(None, 3, 256)	0	conv1d_5[0][0]
concatenate_1 (Concatenate)	(None, 3, 1280)	0	max_pooling1d_1[0][0] max_pooling1d_2[0][0] max_pooling1d_3[0][0] max_pooling1d_4[0][0] max_pooling1d_5[0][0]
flatten_1 (Flatten)	(None, 3840)	0	concatenate_1[0][0]
dropout_1 (Dropout)	(None, 3840)	0	flatten_1[0][0]
dense_1 (Dense)	(None, 2)	7682	dropout_1[0][0]

Figure 2: Composition of TextCNN

3 Evaluation

3.1 Evaluation Metrics

In this section, precision, recall, and micro-average F1 score will be used to evaluate the performance for each model.

3.2 Result

In the section, four models are run with the dev dataset and the test dataset. The result of these four model's performance is shown in the table 1, table 2 and figure 3.

Model	Precision	Recall	F1-Score
Baseline	0.51	0.56	0.53
1ClassSVM	0.6	0.88	0.72
LR	0.87	0.82	0.84
FFNN	0.97	0.86	0.91
TextCNN	0.75	0.84	0.80

Table 1: Result of Dev Dataset

Model	Precision	Recall	F1-Score
Baseline	0.34	0.92	0.50
1ClassSVM	0.38	0.62	0.47
LR	0.56	0.98	0.72
FFNN	0.70	0.92	0.79
TextCNN	0.56	0.90	0.69

Table 2: Result of Test Dataset(Ongoing Evaluation)

Model	Precision	Recall	F1-Score
Baseline	0.34	0.94	0.50
FFNN	0.66	0.83	0.74

Table 3: Result of Test Dataset(Final Evaluation)

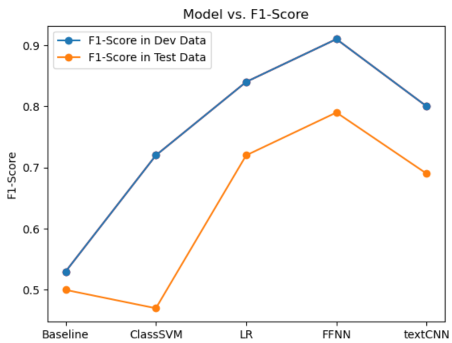

Figure 3: F1-Score of Four Models in Dev Dataset and Test Dataset(Ongoing Evaluation)

3.2.1 Ongoing Evaluation

From the tables and chart, a phenomenon can be seen that one class classifier's performance is much worse than the binary classifier in general. It can reflect that the original data cannot support the OneClassSVM to classify the dev data and test data. After adding negative samples in the train data set, the result improved to another level. Therefore, this model doesn't in the extent of discussion.

Then, focus on the neural network model, figure 4 and figure 5 are the Accuracy and Loss of FFNN and TextCNN, the broken line of FFNN's loss is more smooth than the CNN, also the Val Accuracy of the CNN is low, although the Train Accuracy reach the 1.0, which is possible to overfit.In theoretical, TextCNN model should have better performance than the FFNN and LR, however, the fact is opposite, it's possible that the train set data is not enough and the filter size of CNN is fixed, which limits the horizon of the model. Finally, TextCNN does not perform well than the FFNN model, although it has more complicated procedures.

3.2.2 Final Evalutation

In conclusion, from the data, FFNN model has the best performance among them, therefore, FFNN model is selected as the final choice for the final evaluation, and the result is shown in the table 3. Compared with the baseline, FFNN still has a relatively huge improvement.

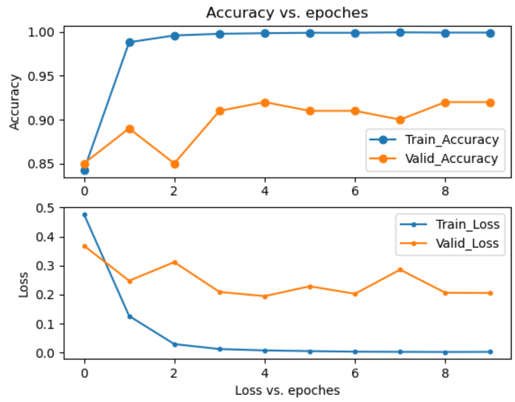

Figure 4: Accuracy and Loss of FFNN

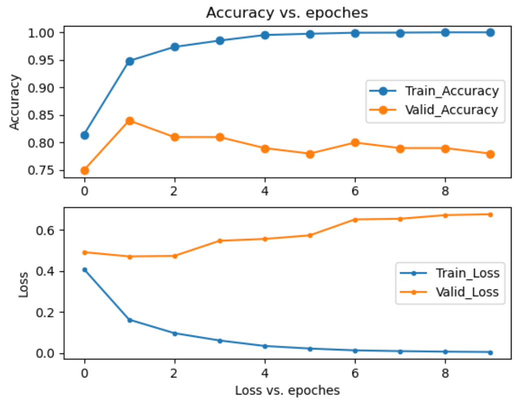

Figure 5: Accuracy and Loss of TextCNN

Model	Negative Sample Misprediction	Positive Sample Misprediction
LR	[15, 36, 54, 63, 70, 92]	[5, 23, 29, 31, 38, 72, 75, 81, 97]
FFNN	[16]	[5, 29, 32, 38, 72, 75]
TextCNN	[3, 8, 14, 15, 18, 36, 54, 67, 68, 70, 71, 78, 96, 98]	[5, 29, 40, 60, 65, 72, 81, 97]

Table 4: Sample Misprediction

4 Error analysis

In this section, the specific prediction result of each model will be discussed and analysed. The following table displays how each model makes prediction mistake. Due to the result of One class SVM is not ideal, this model is excluded into the discussion at this part.
Before analysing, one thing need to be illustrated is that negative sample misprediction means the groundtruth is zero and the prediction is one, the positive sample misprediction is opposite. The specific number of negative sample and positive sample is shown in the table 4.

4.1 Negative Sample Misprediction

From the table, samples from dev set, which is predicted into false result, are shown. In order to evaluate the error, the content of several samples is analysed, in the aspect of the groundtruth is 0 and prediction is 1, the 15^{th} sample, and the 70^{th} sample are selected to analyse. Logistic Regression Model and TextCNN model both make mispredict in the 15^{th} sample and the 70^{th} sample. These two samples are about climate news, the terms, such as 'climate', 'change', are in these samples, which will affect the judgment of the model.From the result, a conclusion can be drawn that, if a negative sample contains enough number of climate terms, it may be recognised as positive.

4.2 Positive Sample Misprediction

Next, turn to the positive sample misprediction, the 5^{th} sample and the 72^{nd} sample are selected to analyse. All of three models mispredict the 5^{th} sample and the 72^{nd} sample, the 5^{th} sample talks about Leonardo DiCaprio Teams Up with Greta Thunberg to Stop Climate Change, but except for the title, there is few 'climate change' in the context, so it possible that this article is recognised as the other types news. However, the 72^{nd} sample contains terms about climate, it still mispredicts, the potential reason is that some political terms, like election and democracy, are in this article, and the negative train data set include political news, which affects the result. Therefore, the number of climate terms affect the prediction result, if a positive sample does not have enough climate terms or contain other terms more than climate terms, it is possible to recognise as negative.

5 Future Work

To improve the result, there are two assumptions for future work.
1. Expanding the train data set. The original train set only contains about 1000 positive samples and without negative samples, which is an unbalanced dataset. However, it is not beneficial to get ideal performance. The dataset should expand with effective negative samples and more positive samples, which can help the model learn more modules of climate misinformation.
2. Extracting more important features for the system, due to the climate truth information and climate misinformation have high similarity, to search the key difference between them is a significant way to classify them.

6 Conclusion

In this report, in order to recognise climate misinformation, a detection system is constructed with pre-process information extraction and multiple machine learning models. The machine learning models, which are OneClassSVM, Logistic Regression, FFNN, TextCNN, have been discussed. After analysing the precision, recall, and F1-score of the dev data set and test data set, as well as the loss line and accuracy line of neural network models, applying FFNN is the temporary best choice for this project. Also, conducting error analysis for the machine learning model with analysing samples for negative sample misprediction and positive sample misprediction, which can help infer the shortcoming of the models. Finally, putting forward two assumptions of future work to improve the result.

References

[1] Rajaraman, A.; Ullman, J.D. (2011). "Data Mining" (PDF). Mining of Massive Datasets. pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.

[2] Vlasveld, R., 2013. Introduction to One-class Support Vector Machines. [Online] Available at: http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/

[3] Tolles, Juliana; Meurer, William J (2016). "Logistic Regression Relating Patient Characteristics to Outcomes". JAMA. 316 (5): 533–4. doi:10.1001/jama.2016.7653. ISSN 0098-7484. OCLC 6823603312. PMID 27483067.

[4] Maheshwari, A.,2018.Report on Text Classification using CNN, RNN & HAN [Online] Available at: https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f

[5] Schmidhuber, Jürgen (2015-01-01). "Deep learning in neural networks: An overview". Neural Networks. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. ISSN 0893-6080. PMID 25462637.