Climate Change Misinformation Detection System

Xinnan Shen
Student ID: 1051380

Abstract

Although we have many ways to get the most accurate information about climate change, there are still many documents telling us the climate change misinformation. It is quite important to detect whether a document contains climate change misinformation to prevent us from being misled from these documents. In this paper, a climate change misinformation detection system is implemented to judge whether some given documents contain misinformation regarding climate change. In the system design and implementation process, some machine learning skills and natural language processing tools are used to detect climate change misinformation based on the contents of documents. The results have shown that the system performs well in predicting climate change misinformation, although some errors still occur.

1 Introduction

These days, climate change has become a big problem that has a significant effect on our life. Some issues caused by climate change such as global warming, the melting of icebergs and the rising of sea levels have triggered serious consequences. Thus, it is quite essential to know accurate information about climate change and do our best to reduce the impact caused by it. However, it is not very easy to distinguish these documents with climate change misinformation from the documents without it just by hand, because their contents look quite similar. Therefore, it is quite useful to design and implement a system to automatically help people distinguish between the documents and tell us whether a given document contains climate change misinformation. In my project, the system is implemented by using machine learning methods and natural language processing tools. The results have shown that the model performs quite well and satisfies the anticipated goals despite the occurrence of some errors.

In this paper, the design and implementation of the system models and the results of them will be deeply discussed. In section 2, I will mention the related work regarding natural language processing and machine learning. In section 3, I will specifically discuss how the system models are designed and implemented. Section 4 will mention the results of each model and section 5 will analyse the possible errors that occur in the model as well as the cause of the errors. In section 6, I will conclude what has been discussed and point out the shortcoming of my system.

2 Related Work

In this following section, I will mention the related researches that have been conducted in this field.

In recent years, natural language processing has developed rapidly and there are quite a few applications of NLP. In Tomek Strzalkowski et al.'s paper (Strzalkowski, 1999), they mentioned that natural language processing can be used for information retrieval so that we can get the main content of documents more quickly. As for neural networks, Qing Ma (Q., 2003), Hang Li (Hang, 2018) and Pranav Kaushik (Sharma Akanksha Rai, 2017) have demonstrated that it is a very powerful tool in many areas of natural language processing. After doing plenty of experiments, they have concluded that the neural network had a pretty good performance and that it had successfully applied to many fields in NLP. In my project, I will also use neural networks as the main tool to detect climate change misinformation in the given documents, and compare the performance of traditional machine learning methods as well.

With respect to the specific method of using

neural networks in natural language processing, many researchers have used feedforward networks to deal with some NLP tasks. Jan A. Botha et al. (Botha Jan A. et al., 2017), Luca Bertinetto (Bertinetto L., 2016) and Martin Sundermeyer et al. (Sundermeyer M., 2015) have argued that many NLP problems are easy to solve with the help of feedforward networks, and that this method is effective in dealing with such problems, like language identification, part-of-speech tagging, although it is far from perfect. In my project, I will compare the performance of feedforward networks with other machine learning models to know whether feedforward networks perform well in this task.

3 System Analysis and Design

In this section, I will mention the procedures of analyzing and designing the system.

3.1 System Analysis

3.1.1 Dataset

The dataset consists of provided dataset and external dataset crawled from the websites, namely NASA [1] and VOA [2]. Specifically, I have downloaded news regarding global climate change from the NASA website and news of some other topics from the VOA website using requests (Reitz et al., 2011). As the data from the NASA website is the truth about climate change, and the data from the VOA website is about topics other than climate change, so these data aren't climate change misinformation. The external dataset is useful for the climate change misinformation detection system, which will be used in the training process. To make the dataset easier to use, the additional data from the website is converted into JSON format, and each document contains a label (1 represents it is climate change misinformation, while 0 represents it isn't).

3.1.2 System Model

In this project, I have used multiple machine learning models to implement the climate change misinformation detection system so that it will be easy to compare the results among different models.

Baseline The baseline of this project is to predict each document randomly. Despite its poor result, the baseline model is quite useful in this project as I can know whether an enhanced model is good enough by looking at the improvements of the F1 score.

Naïve Bayes Before we try to use deep neural networks, we can simply look at how traditional machine learning model performs in this project. For example, we use Naïve Bayes to make predictions. In Naïve Bayes, we assume that each word in a document is generated independently, and the formula of Naïve Bayes (Rennie J., 2003) is shown as below:

$$\hat{y} = \arg\max_{c_j \epsilon C} p(c_j) \prod_{i=1}^{n} p(x_i|c_j)$$

where x_i is the word vector element and c_j is the final label of the given document.

Deep Neural Networks In order to show whether the deep neural networks could perform better than the traditional machine learning tool, Naïve Bayes, I have also used deep neural networks in this project to predict climate change misinformation. As described in section 2, DNN is a very powerful model that can be used to solve many NLP problems. In DNN, each unit calculates the weighted sum of inputs and a bias, and applies a non-linear activation function on the weighted sum to get the output, as is shown in figure 1 below.

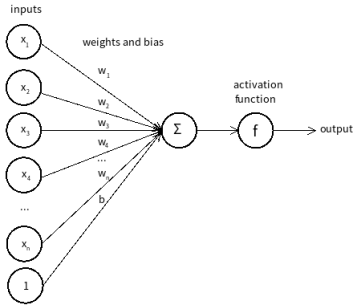

Figure 1: Unit action figure

By using deep neural networks in this project, I can get to know if the model performs better than traditional machine learning models. In particular, I have used the feedforward network in this project. A diagram [3] of feedforward networks is shown in figure 2 below.

[1] Global Climate Change, https://climate.nasa.gov/news/.
[2] Voice of America, https://www.voanews.com.
[3] A Beginner's Guide to Neural Networks and Deep Learning, https://pathmind.com/wiki/neural-network.

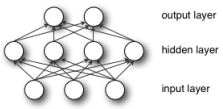

Figure 2: Feedforward networks

The feedforward network consists of three different layers, namely input layer, hidden layer and output layer, and there are normally multiple hidden layers in a feedforward network. In each fully-connected dense layer, each unit performs calculation like figure 1. As is mentioned in section 2, some difficult NLP tasks can be solved with the help of feedforward networks, and I suppose it should be useful in this task as well in distinguishing climate change misinformation.

3.2 System Design and Implementation

The design and implementation of the system consist of three different steps: preprocessing, training and prediction.

3.2.1 Data Preprocessing

After expanding the provided dataset, we need to preprocess the data. First, it is necessary to remove unwanted formats like HTML tags and special characters. Then, I have tokenized the documents into a list of tokens and normalized the tokens into canonical forms. In this step, I have used Part-Of-Speech tagging to lemmatize these tokens. Finally, I have removed unwanted stopwords from the word lists, as they often appear many times in each document and have nothing to do with the document topic. These preprocessing steps mainly utilised NLTK package provided in Python (Bird Steven, 2009), which is a quite popular package in the NLP field. Furthermore, I have also used some tools to convert the bag-of-words representation into feature vectors. The tools are from two packages in Python, sklern (F. et al., 2011) and numpy (Stéfan van der Walt, 2011).

3.2.2 Training

After preprocessing the data, I built the models and trained them using the training dataset.

Naïve Bayes In this step, I have used the package sklearn (F. et al., 2011) to build and train Naïve Bayes. Firstly, I have defined a model of Naïve Bayes. Then, I have conducted hyperparameter tuning. I have tuned the hyperparameter α by using the development dataset. For each α value, I used the training dataset to train the model and calculated the accuracy of development dataset. The best hyperparameter is the one with the greatest accuracy of development dataset.

Deep Neural Networks For DNN, the training process is similar to Naïve Bayes, but the hyperparameters are much more complicated than those of Naïve Bayes. I have used some Python packages in this step, namely tensorflow (Martín Abadi et al., 2015) and keras (François et al., 2015). As this project is a binary classification task, I used sigmoid as the activation function and set 1 unit in the output layer. In addition, the loss function should be "binary cross-entropy" because of that. Meanwhile, I have treated the number of units in hidden layers and the optimizer function as hyperparameters and tuned them by using development dataset. After that, I could finally find a set of hyperparameters with the best performance in development dataset. However, I have also noticed that the number of hidden layers has an effect on the performance of the model. So I have built three models with 2, 3 and 4 hidden layers and compare their performance in the development dataset.

3.2.3 Prediction

After training the model and tuning the hyperparameters, I could select the best hyperparameter with the greatest accuracy in development dataset. Thus, by using the models with the best hyperparameters, it is very likely that the system can predict great result. Finally, I have used test dataset to make predictions and save the prediction results as the format of JSON.

4 Results

In this section, I will show the result of each model as well as analyze the results.

4.1 Baseline

Table 1 shows the results of baseline. It is evident that the performance of the baseline model is not quite good, and the model can be enhanced to get a better result.

Table 1: Baseline Result

Precision	Recall	F1 Score
0.519	0.560	0.538

4.2 Naïve Bayes

When using the Naïve Bayes model, the performance of the climate change misinformation detection system has improved, and the results can be seen in table 2 below. However, the performance can still be improved.

Table 2: Naïve Bayes Result

Precision	Recall	F1 Score
0.686	0.960	0.800

4.3 Deep Neural Networks

In terms of deep neural networks, I have used three different models with 2, 3 and 4 hidden layers respectively. The results for these models can be seen as figure 3.

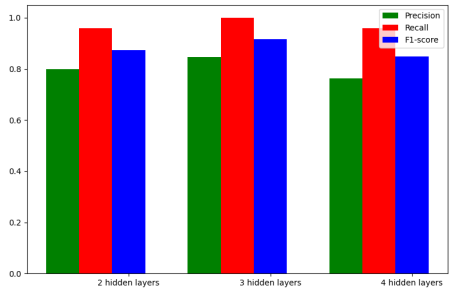

Figure 3: Deep Neural Networks Result

From the graph, we can see that the performance of feedforward networks is better than Naïve Bayes in this particular task, and the deep neural networks with 3 hidden layers have the best performance, with F1 score 0.917.

5 Error Analysis

In this section, we present error analysis for the models that have been used in the system.

5.1 Naïve Bayes

When using Naïve Bayes, the system performs better than random baseline, but it has a lot of shortcomings. First of all, it is based on the conditional independence assumption. But in reality, the occurrence of each word in a document is not independent of each other at all. In addition, it makes predictions based on the probability calculation, which is based on the bag-of-words representation for each document. In fact, whether a document contains climate change misinformation only depends on its content and topic, and some words are not related to the topic, which could have led to the mislabelling.

5.2 Deep Neural Networks

Despite neural networks have great performance in this task, it is not perfect at all. The feedforward networks have mislabelled some documents, because we only care about the number of times a word appears in a document and treat it as an element in the document vector. In fact, some documents contain many words that indicate the documents are likely to contain climate change misinformation, but the topic is not about climate change misinformation. For example, in one document, it repeatedly mentioned *carbon dioxide* and *fossil fuels*, but it mainly talks about the economy rather than climate change. Also, only caring about the occurrence time of each word in a document is not enough, as some words only represent meaning related to climate change when they appear together, like *global warming*. If we can feed the network with the vector representing the contents rather than the word occurrence, the performance of the system could be further improved.

6 Conclusion and Further Improvement

6.1 Conclusion

At present, there are many documents containing climate change misinformation. In this project, I have designed a system to classify documents. Although the system has some shortcomings, it performs quite well and can achieve estimated goals.

6.2 Further Improvements

There are many aspects to consider so that the performance of the system can be improved. Firstly, I have only considered DNN with 2, 3 and 4 hidden layers. It is worth considering a model with more than 4 hidden layers as well. In addition, the model has only used the one-hot vector. The performance of the model could be better if we use word embeddings and LSTM. As stated in section 5.2, there are some problems if we only treat the number of word occurrence as input vectors. Furthermore, it is better to make comparisons between different neural networks in this problem apart from DNN, such as CNN and RNN.

References

Valmadre J. Torr P. Vedaldi A. Bertinetto L., Henriques J. F. 2016. Learning feed-forward one-shot learners. *Advances in neural information processing systems*, pages 523–531.

Loper Edward Bird Steven, Klein Ewan. 2009. Natural language processing with python: [analyzing text with the natural language toolkit].

Pitler Emily Botha Jan A. et al. 2017. Natural language processing with small feed-forward networks.

Pedregosa F., Gramfort A. Varoquaux G., et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Chollet François et al. 2015. Keras. `https://github.com/fchollet/keras`.

Li Hang. 2018. Deep learning for natural language processing: advantages and challenges. *National Science Review*, 005(1):24–26.

Paul Barham Martín Abadi, Ashish Agarwal et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ma Q. 2003. Natural language processing with neural networks. In *Language Engineering Conference, 2002. Proceedings*.

Ken Reitz et al. 2011. Requests.

Teevan J. Karger D. Rennie J., Shih L. 2003. Tackling the poor assumptions of naive bayes classifiers. *ICML*.

Kaushik Pranav Sharma Akanksha Rai. 2017. Literature survey of statistical, deep and reinforcement learning in natural language processing. In *International Conference on Computing*.

Tomek Strzalkowski. 1999. *Natural Language Information Retrieval. [electronic resource].* Text, Speech and Language Technology: 7. Springer Netherlands.

Schluter R. Sundermeyer M., Ney H. 2015. From feedforward to recurrent lstm neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):517–529.

Gaël Varoquaux Stéfan van der Walt, S. Chris Colbert. 2011. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13:22–30.