COMP90042 Project 2020: Climate Change Misinformation Detection

Jiayu Xia
947875

Abstract

The purpose of this article is to show the process of selecting and building a system which detects whether a document contains climate change misinformation or not. Since the training datasets are only with positive labels, the approaches we take to expand training datasets is to crawl documents with negative labels and then apply binary classification techniques to classify the development datasets and test datasets. The main classification techniques that we applied is Logistic Regression (Kleinbaum et al.,2002) and Multilayer Perceptron (MLP) (Pal et al.,1992). The features I chose is TFIDF vectors. Other techniques like Support Vector Machines (SVMs) (Schölkopf et al., 2002) is also considered during the process of improving the performance of prediction. The results indicate that MLP performed better in predicting the test instances while Logistic Regression had a high evaluation accuracy on the development set.

1 Introduction

In recent years, the amount of misinformation on the internet has increased significantly. As a result, fact-checking has become increasingly important in our life. In terms of a widely discussed topic, climate change has become world focus of attention. Although climate change is an inevitable reality, there are lots of articles containing climate change misinformation such as fear-mongering and science-discrediting behind climate change. Thus, this project aims at distinguishing climate change misinformation.

In order to deal with the supervised binary classification problems and evaluate the predicted results, five processes are taken, including document acquiring, pre-processing, information retrieval, model application and method evaluation.

Document acquiring: Crawl data with negative labels: the data without climate change misinformation or data not related to climate change.

Pre-processing: The data cleaning process includes removing HTML, turning the text into lowercase, word tokenisation, remove punctuations and numbers, removing stopwords in the tokens and using Porter stemmer to remove the commoner morphological and inflexional endings from words in English.

Information retrieval: Using the TFIDF vectoriser to retrieve the information of a document and turning it into features (Jing et al., 2002).

Model application: Apply supervised binary classification methods, including Multilayer perceptron and Logistic Regression to predict the data.

Method evaluation: Evaluate the methods using the development data.

According to the ongoing evaluation on Codalab, the best F1 score based on the test set is 0.72.

2 Background

In this section, the underlying methods that I adopted for the development of the system are presented.

2.1 Website crawling

Website crawling is the automated fetching of web pages by a software process, and the purpose is to get the information on the website for analysing.

2.2 Multilayer perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Each node

is a neuron which uses nonlinear activation function except for the input layer. What's more, MLP uses a supervised learning method called backpropagation for training.

2.3 Logistic Regression

The logistic regression model is used for modelling the probability of a specific class or event in terms of input and does not perform classification function. However, it can be used to make a classifier by choosing an appropriate cutoff value, and we classify the input into one class with the probability high than the cutoff value into one class and below the cutoff value into another.

2.4 Support Vector Machine

Support vector machines (SVMS) are a set of supervised learning methods used for classification, regression and outlier detection.

3. Methodology

3.1 Crawling data for negative labels

As explained in section 2.1, I implemented a crawling method to acquire data from www.theguardian.com, which is a British news and media website owned by the Guardian Media Group. This website covers subjects of all kinds, and since this website only contains information of reality, I crawled data and labelled them as '0'. The data I crawled includes climate change information, sports news, film, games, music, book, life and style, global development and business. The crawled data are added into trainset with labels 0. After that, all the data are pre-processed, as mentioned in section 1.

3.2 Feature selection

After the data had been pre-processed into tokens, I selected bag-of-words (BOW) representation as features, which describes the occurrences of words within a document (Zhang et al., 2010). However, there are shortages in BOW, such as the rareness of a term is not considered. Then I considered using TF-IDF representation as features to test the accuracy to overcome the drawbacks of BOW because it can find out how important a word is to a document in a collection, especially for rare words. TF-IDF = Term Frequency (TF) * Inverse Document Frequency (IDF). TF is counts of tokens in a document while IDF is used to diminish the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. With the usage of TF-IDF representation, we can give importance to the rarity of a word. For instance, the word 'climate' is sure to be common among all documents, and the word with less regularity like 'disgraceful' in the example climate change misinformation should be laid more emphasis on. In the project, I focused on the word tokens with term frequency more than 25 and used them as features to the classification models.

3.3 Model selection

Firstly, I experimented on famous Supervised Machine Learning methods like SVMs and Logistic Regression. Logistic regression shows better performance on the development set and the weighted average F1 score of which is 0.86. By contrast, the weighted average of LinearSVC is 0.83. However, when I uploaded the predicted labels on the test set on Codalab, the F1 score of Logistic Regression and SVM was 0.6929 and 0.6903 relatively on positive class, which is lower than their results on the development set. Then I tried to build a deep learning model to improve the predicting accuracy on the test set. Under this circumstance, I used the feedforward neural network (Multilayer Perceptron) to takes in feature inputs. The inputs are then processed in hidden layers using weights that are adjusted during training. Finally, I got a prediction accuracy as the output of the model. Though the F1 score on development set was 0.82, the F1 score on positive class on the test set was improved to 0.72.

4 Results

Table 1 shows the performance of Logistic Regression, MLP and SVMs on the development set. Here we provide the precision, recall and F1-score for each class. The result indicates that based on the development data set, Logistic Regression classifier outperforms LinearSVC classifier and LinearSVC classifier outperforms MLP classifier. However, in terms of the test data set, the F1-score of logistic Regression, MLP and LinearSVC are 0.6929,0.7193 and 0.6903, relatively, which indicates that MLP performs better than the other two classifiers on the test dataset.

Performance on development data set	*Logistic Regression+TF-IDF features*	*Multilayer Perceptron+TF-IDF features*	*LinearSVC classifier +TF-IDF features*
Precision for class '0'	0.81	0.74	0.76
Precision for class '1'	0.93	0.97	0.95
Recall for class '0'	0.94	0.98	0.96
Recall for class '1'	0.78	0.66	0.70
F1-score for class '0'	0.87	0.84	0.85
F1-score for class '1'	0.85	0.79	0.80
Macro average	0.86	0.82	0.83
Weighted average	0.86	0.82	0.83

Table 1: Performance of Logistic Regression, MLP and LinearSVC classifiers

In transforming TF-IDF vectors, the parameter max_features I chose is 4353 at the beginning, which is the number of word tokens that exists more than 25 times in the datasets. The reason for doing this is that some exceptionally rare words will add unwanted dimensions to inputs. Then I wanted to discover how the number of Max_features influence the F1 score. With the usage of Pyplot figure, Figure 1 and Figure 2 show the influence of parameter max_features on Logistic Regression and MLP.

From the two figures, we can find that the precision, recall and F1 score remain stable when max_features reach 6000, which proves that it does not make improvements to add some rare words as features. In general, the F1 score of Logistic Regression improved slowly with the increase of max_features until the point (6000) that adding features by rare words does not make sense to the prediction. By contrast, the F1 score of MLP reached highest at 0.831 when max _features was 500. Besides, SVMs show the best performance on precision (1.0), recall (0.74) and F1 score (0.85) when the parameter max_features is equal to 1500.

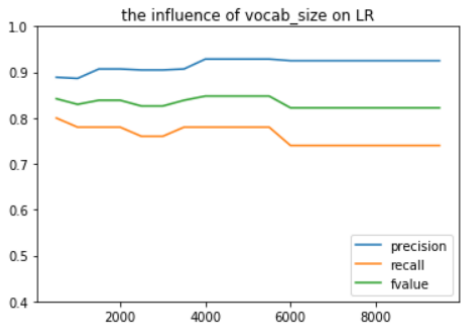

Figure 1: the influence of max_features on Logistic Regression

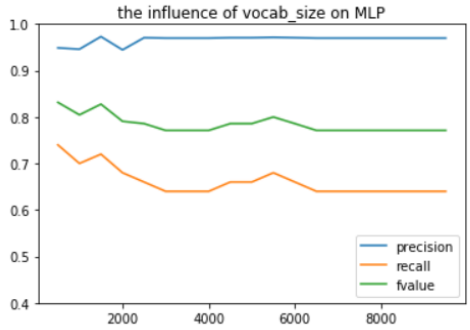

Figure 2: the influence of max_features on Multilayer Perceptron

5. Error analysis

Results from Table 1 shows the precision, recall and f1 score of different classifiers on predicting climate change misinformation. Precision is defined as the number of true positives over the number of true positives plus the number of false positives. Recall is defined as the number of true positives over the number of true positives plus the number of false negatives. In general, the precision of

positive class is much higher than that of negative class, which indicates that the classifiers had a low false positive rate on positive class. Besides, the recall of negative class is much higher than that of positive class but the precision is lower, which indicates that the classifiers had a low false negative rate on negative class and the system returned many results on label 0, but many of its predicted labels were incorrect when compared to the training label, and thus, the system made more errors on negative class.

The reason why we made more errors on predicting the instances with label 0 is that the data with label 0 are those we crawled from the website and those data are not so representative for the classifier to predict. In the beginning, I added only the genuine climate change information in Train dataset, and the f1 score was only around 0.7. In order to achieve higher performance, I looked at the style of dev data to see what kind of topics it contains. After adding data related to sports and other topics that should also be put in negative class, the precision and f1 score on negative class increased, which was proved to be a method for reducing errors on negative class.

Also, the proportion of data with label 0 v.s label 1 is 1:1 in dev dataset, and I made a hypothesis that the number of data with negative labels in train set might influence the precision, recall and F1 score. Figure 3 shows the influence of negative set size on the precision, recall, and f1 score on positive class. The number of instances in positive class in train set is 1168, and I limited the length of the negative set from 1168 to 4088. In terms of Logistic Regression classifier, the results indicate that with the increase of negative set size, the precision and recall of class 1 increased as well. However, in terms of MLP in Figure 4, with the increase of negative set size, the precision score of class 1 increase while the recall score of class 1 decrease, the highest f1 score was reached when the size of the negative set is approximately equal to the size of the positive set. Besides, SVMs made fewest errors, with precision(0.93), recall(0.80), F1 score (0.86) on class 1 when the size of negative set is 1752.

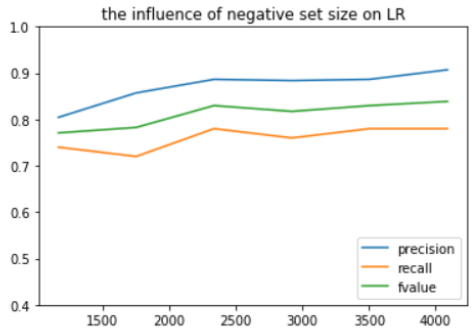

Figure 3: The influence of negative set size on Logistic Regression

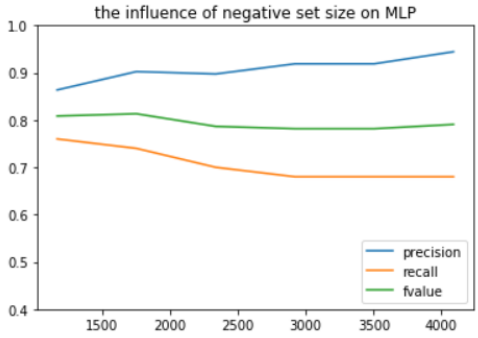

Figure 4: The influence of negative set size on MLP

6. Conclusion and implication

In this paper, I present the system using SVMs, MLP, and Logistic Regression classifiers to predict the climate change misinformation problem. According to the ongoing evaluation on Codalab, the highest F1 score on positive class on the test dataset is 0.72 by MLP, while the highest f1 score on positive class on the development dataset is 0.86 by Logistic Regression. In terms of the final evaluation on Codalab, the best F1 score is 0.68. To further this study and improve performance in future design, I suppose using Elmo embeddings which are learned from Bidirectional Long Short Term Memory (BiLSTM) (Chen et al., 2017; Peters et al., 2018). BiLSTM is good to be used on occasions when the learning problem is sequential. Also, BiLSTM knows when to keep and forget using gates in their architecture. Elmo embeddings are learned from the internal state of a BiLSTM and represent contextual features of the input text. This feature overcomes the shortage that TFIDF vectors do not consider the order of a sentence. In brief, I would use Elmo embeddings to improve the performance in future design.

Reference

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). Logistic regression. New York: Springer-Verlag.

Pal, S. K., & Mitra, S. (1992). Multilayer perceptron, fuzzy sets, classifiaction.

Schölkopf, B., Smola, A. J., & Bach, F. (2002). Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics, 1(1-4), 43-52.

Jing, L. P., Huang, H. K., & Shi, H. B. (2002, November). Improved feature selection approach TFIDF in text mining. In Proceedings. International Conference on Machine Learning and Cybernetics (Vol. 2, pp. 944-946). IEEE.

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. Expert Systems with Applications, 72, 221-230.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.