Information Retrieval Coursework Report

Zonghan Zhang
University College London
School of Computer Science
ucabzz4@ucl.ac.uk

1 Introduction

Stance detection has been a very important topic in Natural Language Processing and Information Retrieval field. One of the key problem in stance detection is to identify fake news, which is what we need to do in this coursework. The given FNC-1 data set contain body article and headline with four different stances, including 'agree', 'disagree', 'discuss' and 'unrelated'. The main work of the coursework is to find important features from the data set, such as vector representation, language model based representation, and then implement simple linear models to predict and explore some state-of-the-art models and features.

2 Solution

2.1 Task1: Training and validation split

From the description of the data set we can see that the data set is highly imbalanced that the 'unrelated' headline is the most frequent class in the data set. In order to keep the 'imbalanced' in the sub training set and validation set, I splitting the data set on class level. The final statics of the ratio is give in Table 1.

Table 1: Statics ratio of taks 1

data set	agree	disagree	discuss	unrelated
train	3678	840	8909	36545
sub train	3311	756	8019	32891
validation.	367	84	890	3654

2.2 Preprocess

Before giving the description of other sub tasks, I would first give short introduction to my methods to preprocess the data, which is very important in the coursework and also the foundation of other sub tasks. For the articles of body and the headlines of the stance, I first merged the training and test set before the process. I used the self-custom tokenization function to split the data into words and then lowered all the characters. After these processes, I removed the stop words, such as a and the, and then performed lemmatization. Then, a very important process is to establish a word corpus dictionary which records all possible words in the data set and also their corresponding indexes. Finally I transformed all the word in the data to numbers, which are index of the work. This makes it better to deal the data and complete other sub tasks.

2.3 Task2: Vector Representation

In this task, I extract four different representations of the article and headline and compute the cosine similarity of these representation to find out which one is meaningful in the data set. For word2vec based representation. I use the pre-train word2vec data set given in lecture9.

2.3.1 *Salton's vector space*

The Salton's vector space is actually the bag-of-word method. The difference is that the vector of a document is represented by the TF and IDF value of all the word in the word corpus. Then cosine similarity is used to measure the similarity.

2.3.2 *Averaga Word2vec Representation*

Based of the word2vec representation, we can average all the words in a document to get the final vector of a document. But I make some changes of the method. What I did is to first multiply the word vector by their IDF values and then average them. The idea is that the word with higher idf value should be placed more importance. For this method, cosine similarity is used.

2.3.3 *Word2vec and TF-IDF based Representation*

For this representation, I use the TF-IDF values to selct n most important word as the topic words in the documents, where n is a parameter of this method. Then I use the word mover's distance to measure distance between articles and headlines. This distance will be used as a feature in the final models.

2.3.4 *Doc2vec Rpresentation*

It is also possible to train the doc2vec model using the Genism package. The doc2vec model is trained using all the article and headline in training set and test set. Then, cosine similarity between article and headline is used to measure similarity.

2.4 Task3: Language Model

The Smoothing method used in this task is a combination of Discounting method and Interpolation method. I first get the background probability of body and headline respectively and then apply the Dirchelet Smoothing. The parameter u is set as the average length of the document. Then the already computed tf-idf matrix can be used to get word probability efficiently. But there is another problem. Word in headline background may not appear in the article background. So I also apply the Discount method is add a very small value to all words to avoid zero probability.

2.5 Task4:Alternative Feature

In this task, I extract two topic model feature of the articles and the headlines. The first one is the LSA(latent semantic analysis) feature. What I did is to first apply SVD on the term frequency matrix as Figure 1 shows. Then select K most important topics from the data set. Then I multiple the diagonal matrix by the C matrix which finally gives me the (K, D) matrix. The matrix explains how related are the documents to the topics. The related headlines and articles should have the related topics. Therefore we can use the cosine similarity of the article vector and headline vector to measure their similarity. Another popular topic feature model is LDA(Latent Dirichlet Allocation) which is implemented based on the SVD method. I implement this method by using the sklearn package and also use the cosine similarity to measure the feature important.

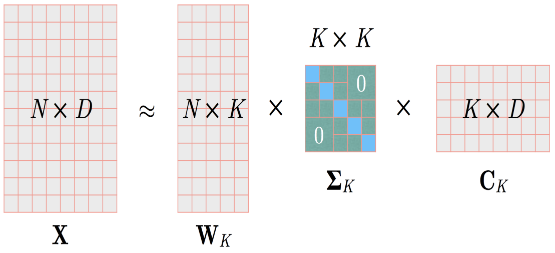

Figure 1: Latent Semantic Analytic

2.6 feature importance

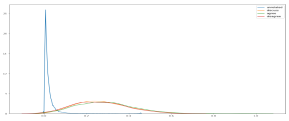

Figure 3: Salton' vector space cosine similarity distribution

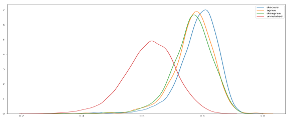

Figure 4: Average word2vec method cosine similarity distribution

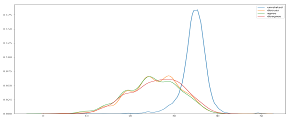

Figure 5: Average word2vec method word mover's distance distribution

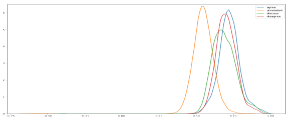

Figure 6: Doc2vec method word cosine similarity distribution

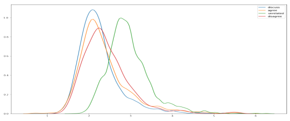

Figure 7: Language model KL divergance distribution

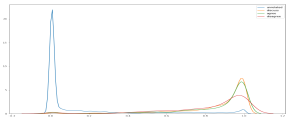

Figure 8: Topic model cosine similarity distribution

Figure 3-8 show the distributions of cosine similarity or other distance between articles and headlines for four different stances. From the figures we can see that all the distances are not able to help distinguish among 'agree', 'disagree' and 'discuss'. However, we can clearly see that the Salton's vector representation and LDA model can distinguish related headlines from unrelated headlines. Therefore, the Salton's vector representation and the LDA model based representation are the most important features(representations) in the task. A surprising finding is that the word2vec methods do not work well in this task. We can see from figure4 that the majority of the distance are close to one, which is hard to make good prediction on the label. Possible reasons are that our tokenization function is not good enough and I should train my own topic word2vec model for this task.

2.7 Linear Model

In this task, I implement two linear models, including linear regression and logistic regression to do the classification problem. In additional to use gradient descent as the optimization method, I also implemented the batch gradient descent and the L2 regularization to avoid overfitting.

2.7.1 *Linear Regression*

The loss function used in linear regression is the mean square error. The linear regression is not commonly used in the binary classification problem because the predict is not range in 0 to 1. But we can still predict positive if the prediction is larger than 0.5 and negative if the value is less than 0.5. Another problem is that this is a multi-classification problem. So I would train four linear regression models and make prediction based on the highest predicted value.

2.7.2 *Logistic Regression*

The Logistic Regression is more common in the binary classification problem. Since our task is multi-classification problem. I implement the one-vs-all algorithm. For multi-class problem, I train a binary logistic regression model for every class. Once I get the probability prediction of all classes, I apply a softmax function which gives prediction where all probabilities sum up to one. The model would make prediction based on the softmax prediction.

2.7.3 *Performance Evaluation*

As we know that the data set is highly imbalanced. Therefore the overall accuracy is not enough to measure the performance of the model. If we just predict unrelated for all sample, we can still get a "good" accuracy of the data set. Another good metric is the Accuracy Under Curve. The idea of AUC is similar to precision-recall and it computes the true positive rate and false positive rate to measure the performance. Besides, it can handle the imbalanced data and give explainable result of the model. The final metric I used in the task is the confusion matrix, which would give detailed information of our performance and give suggestions on how to improve our model.

One important thing before training the model is to select the vector representation of article and the headline. We can see form figure3 and figure 8 that both the BOW and Topic model performs well on the data. The topic model is trained using the BOW representation. Therefore, topic model representation are more suitable for this problem. The final selected features are presented in Table 2.

Table 2: Final features

name	description	dimension
feature1	article topic model vector	25
feature2	headline topice mode vector	25
feature3	Salton's vectore cosine similarity	1
feature4	average word2vec cosine similarity	1
feature5	word mover's distance	1
feature6	doc2vec cosine similarity	1
feature7	LDA model cosine similarity	1
total		55

Table 3: Confusion matrix of Linear Regression

	Agree	Disagree	Discuss	Unre.	Acc.
Agree	180	0	120	1603	9.46%
Disagree	49	0	32	616	0%
Discuss	196	0	366	3902	8.20%
Unre.	342	0	1	18806	98.13%
total					73.00%

Table 4: Confusion matrix of Logsistic Regression

	Agree	Disagree	Discuss	Unre.	Acc.
Agree	730	296	375	502	38.36%
Disagree	151	123	48	375	17.64%
Discuss	1348	586	1344	1186	30.11%
Unre.	15	20	11	18303	99.75%
total					80.67%

Table 5: AUC of two linear models

Model	AUC
Linear Regression	0.5324%
Logistic Regression	0.6806%

The two models are trained using different parameters to get better performance. Due the page limit, I would not give the description of the parameters in this report. Table 3 and Table 4 give the confusion matrix as well as the accuracy of each stance of linear regression model and logistic regression model. Table 5 gives the AUC score of two models. It is clear that the logistic regression performs better than linear regression on classification problem. Although they both did a good job on classifying the unrelated headline, the linear regression failed to predict the disagree class, which is unsatisfactory. From the confusion matrix we can see that the it is still very difficult for the models to distinguish among 'agree', 'disagree' and 'discuss'. There is a trade-off between using large learning rate or small learning rate. A large learning rate means a shorter training time because the gradient declines much fast at each iteration but it is hard to converge. If we choose a small learning rate, it would take longer time to converge and it also may not converge if the value is too small. Therefore, the learning rate is an important parameter in the model.

Table 6: model Accuracy after dropping feature

	AUC
No drop	0.8067%
feature3(tf-idf cosine similarity)	0.8067%
feature4(word2vec cosine similarity)	0.8066%
feature5(word mover's distance)	0.8051%
feature6(doc2vec cosine similarity)	0.8079%
feature7(topic model cosine similarity)	0.8060%

To evaluate the importance of distance feature, I train models by dropping a specify feature and compare their performance. The result is given in Table 6. We can see that after dropping the word mover's distance, the overall accuracy dropped the most, which may indicate that the word mover's distance might the most important feature in the vector. Also, the doc2vec cosine similarity seems to have negative impact on the model.

3 Literature Review

There are many methods that achieve good performance on the FNC1 competition. As we know that Recurrent Neural Network like LSTM, BiLSTM have been standard methods to solve NLP problems. However, many teams pointed out that the RNN may not be suitable for the task. And some people found out that the multi-layer perceptron did really good. Also, two of the top three teams applied such method while the winner team apply ensemble methods by combining Convolution Neural Network and Gradient Boosting Machine to win the competition. To sum up, deep model like MLP and CNN got good performance. In additional to the model, feature engineering is also crucial part of the task. The most popular representation of body and headline is the BOW representation and the top three teams used this feature in their models. The different is that some applied different n-gram model while other teams applied unigram model. Other features including count features, topic model features, TF-IDF features and word2vec features are widely used in the competition. Surprisingly, few teams employed doc2vec model in the competition. In general, count feature like number of overlaping words between headline and article, n-gram features with corresponding TF-IDF values and the topic model features play important roles in the stance detection task.

4 Deep Model

In this part, I employ some popular python packages like sklearn and keras to extract new features and implement deep model in the stance detection task. Although the bow representation seems useful according to the final reports of many teams. The topic model representation works better than the bow representation in my model. Besides, the low dimensions of the topic model also accelerate the training process. In order to solve the imbalanced problem, I apply down-sample on unrelated data, which reduces the size of the training set. However, this method seemed to have negative impact on the neural network. So I gave up sampling method in the part. Table 7 shows my final features. Table 8 shows the architecture of the multi-layer perceptron.

Table 7: Final features

name	description	dimension
feature1	article LDA topic model vector	25
feature2	headline LDA topice mode vector	25
feature3	Salton's vectore cosine similarity	1
feature4	average word2vec cosine similarity	1
feature5	word mover's distance	1
feature6	doc2vec cosine similarity	1
feature7	LDA model cosine similarity	1
total		55

Table 8: Network Architecture

Layer	Output Shape
Input	55
Dense	128
Activation(relu)	128
Dropout(0.6)	128
Dense	64
Activation(relu)	64
Dropout(0.5)	64
Dense	4
Activation(softmax)	4

Table 9: Confusion matrix of final model

	Agree	Disagree	Discuss	Unre.	Acc.
Agree	141	0	1626	136	7.41%
Disagree	71	0	490	136	0%
Discuss	189	0	3980	295	89.16%
Unre.	20	0	347	17982	98.00%
total					87.00%

From Table 9 we can see that the deep model performs much better than the logistic regression. It works better on classifying the unrelated headline as well as the discuss headline. But it do worse than logistic regression in classifying the agree headline and disagree headline. It is worthy to mention that the deep model have 97% accuracy on classifying the related headline but it still fails to predict agree and disagree headline.

5 Conclusion

The coursework has taught me many important lessons in Information Retrieval, Natural Language Processing and Machine Learning. By implementing algorithms on my own instead of employing existing packages, I have a deeper understanding of state-of-the-art algorithms and the ideas behind them. Besides, the stance detection task is very interesting and challenging. There are many things and methods that can make improvement we can try in the coursework. It also means that I need to keep studying so as to solve practical problems in the real world.