Work summary of 14th week

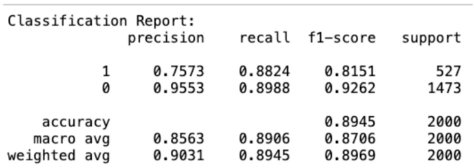
Student: Xuanyu Su
Supervisor: Isar Nejadgholi

December 2, 2020

1 BERT model construction with EA(East Asian prejudice) data without fine-tuning

In view of the fact that in the previous work, the effect of training and prediction on EA data is the best, hence in this module, we use EA data as training data to train the BERT and RoBERTa models, and perform the 2400 Covid data respectively. The test results are as follows.

1.1 Model test on EA data

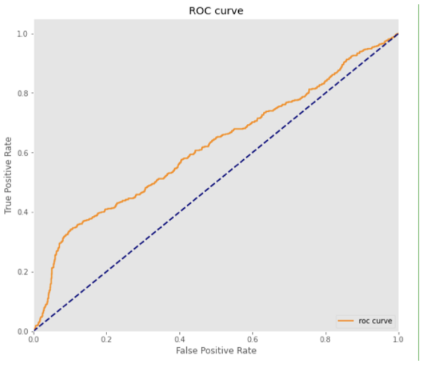


```
Classification Report:
              precision    recall  f1-score   support

           1     0.7573    0.8824    0.8151       527
           0     0.9553    0.8988    0.9262      1473

    accuracy                         0.8945      2000
   macro avg     0.8563    0.8906    0.8706      2000
weighted avg     0.9031    0.8945    0.8969      2000
```

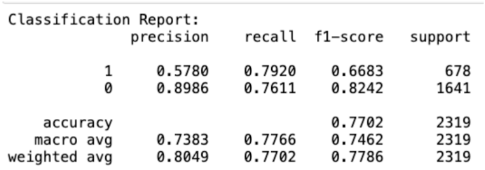
The classification report of BERT model test on EA data



The ROC curve of BERT model test on EA data

Figure 1: The classification report and ROC curve of BERT model test on EA data

1.2 Model test on 2400 Covid data

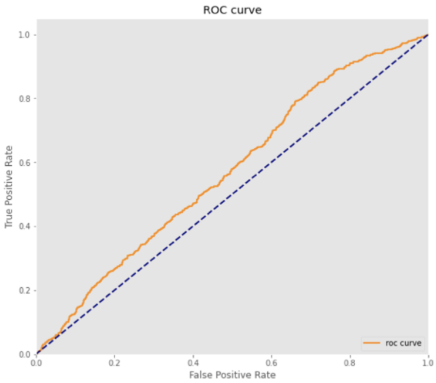


```
Classification Report:
              precision    recall  f1-score   support

           1     0.5780    0.7920    0.6683       678
           0     0.8986    0.7611    0.8242      1641

    accuracy                         0.7702      2319
   macro avg     0.7383    0.7766    0.7462      2319
weighted avg     0.8049    0.7702    0.7786      2319
```

The classification report of BERT model test on 2400 Covid data

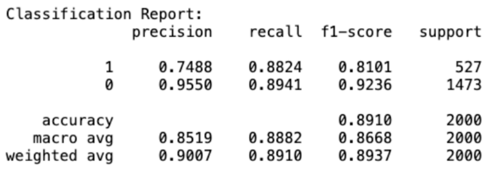


The ROC curve of BERT model test on 2400 Covid data

Figure 2: The classification report and ROC curve of BERT model test on 2400 Covid data

2 RoBERTa model construction with EA data without fine-tuning

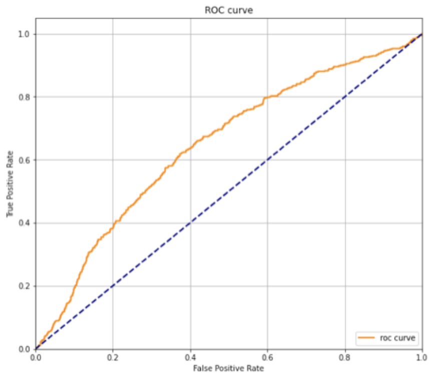
2.1 Model test on EA data



```
Classification Report:
              precision    recall  f1-score   support

           1     0.7488    0.8824    0.8101       527
           0     0.9550    0.8941    0.9236      1473

    accuracy                         0.8910      2000
   macro avg     0.8519    0.8882    0.8668      2000
weighted avg     0.9007    0.8910    0.8937      2000
```

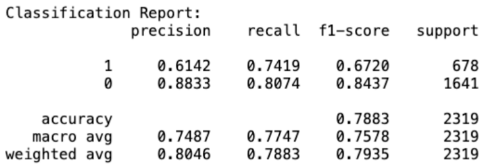


The classification report of RoBERTa model test on EA data

The ROC curve of RoBERTa model test on EA data

Figure 3: The classification report and ROC curve of RoBERTa model test on EA data

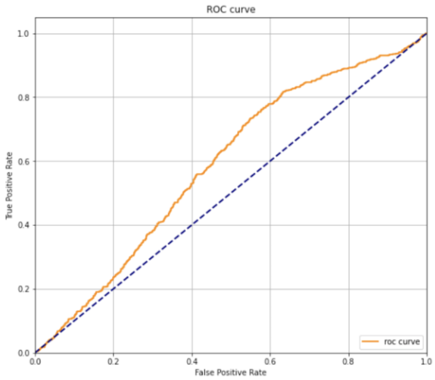
2.2 Model test on 2400 Covid data



```
Classification Report:
              precision    recall  f1-score   support

           1     0.6142    0.7419    0.6720       678
           0     0.8833    0.8074    0.8437      1641

    accuracy                         0.7883      2319
   macro avg     0.7487    0.7747    0.7578      2319
weighted avg     0.8046    0.7883    0.7935      2319
```



The classification report of RoBERTa model test on 2400 Covid data

The ROC curve of RoBERTa model test on 2400 Covid data

Figure 4: The classification report and ROC curve of RoBERTa model test on 2400 Covid data

2.3 Summary

Through the observation of the above results, we found that the BERT model and RoBERTa model trained with EA data have achieved better performance on the 2400 Covid data. The RoBERTa model trained with EA has the best results when testing on 2400 Covid data, which could achieve the F1 score with 0.6720 for positive class, and 0.8437 for negative class. This also indicate that more detailed classification of data and hashtags could make the model more accurate identification and classification.

3 Modification on Distance Comparison graphs

After the experiment last week, we found from figure 5 that in the distance comparison graph without fine-tuning operation in the linear layers, the distribution difference among the databases is not obvious. Therefore, in this week's experiment, we modified the data distribution graph to make the distribution of data more obvious for each individual dataset. The specific results are as follows.

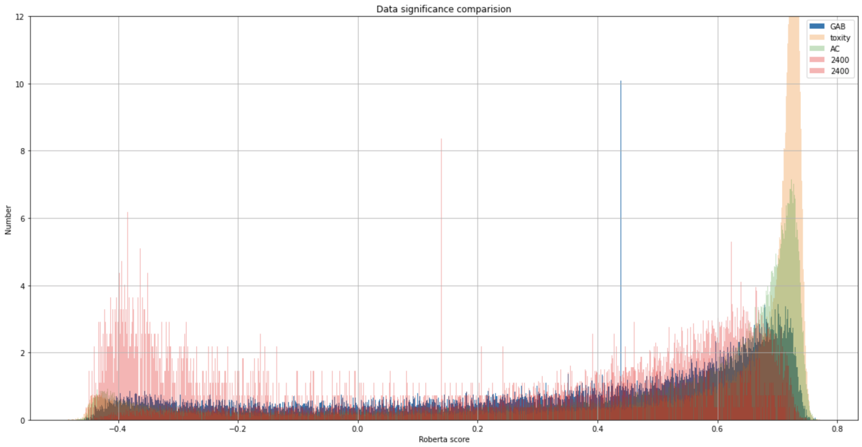


Figure 5: The distance comparison graph of five databases test on the RoBERTa model trained with Toxicity Data

The graph after we apply Gaussian distribution as shown in the below:

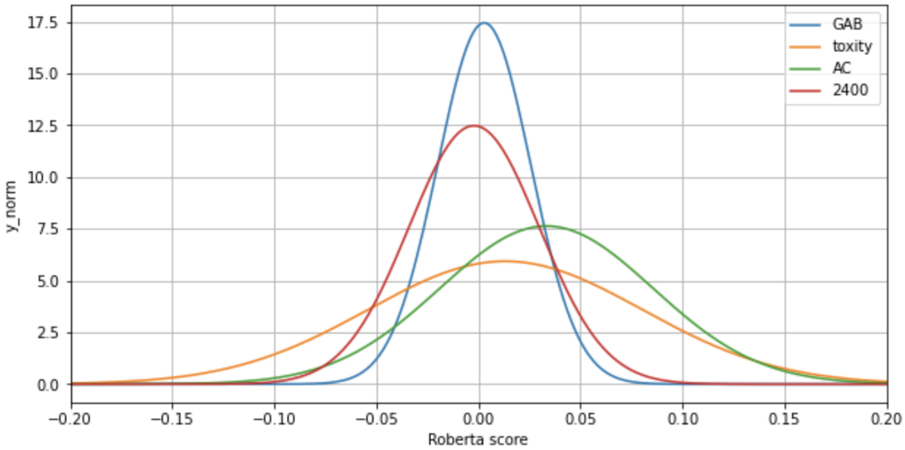
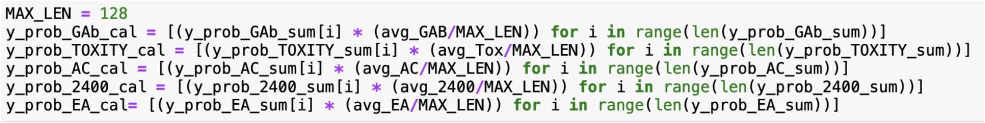


Figure 6: The distance comparison graph of five databases test on the RoBERTa model trained with Toxicity Data after modification

3.1 Changes made in the new version of distance comparison

In this version of distance comparison, we made the following changes:

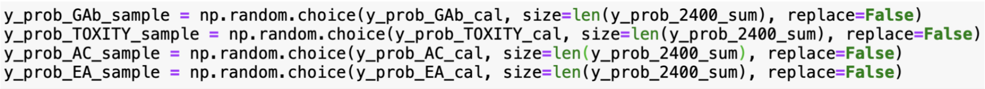
1. calculate the normalized sentence length for each database, and multiply with the sum of RoBERTa score.



```
MAX_LEN = 128
y_prob_GAb_cal = [(y_prob_GAb_sum[i] * (avg_GAB/MAX_LEN)) for i in range(len(y_prob_GAb_sum))]
y_prob_TOXITY_cal = [(y_prob_TOXITY_sum[i] * (avg_Tox/MAX_LEN)) for i in range(len(y_prob_TOXITY_sum))]
y_prob_AC_cal = [(y_prob_AC_sum[i] * (avg_AC/MAX_LEN)) for i in range(len(y_prob_AC_sum))]
y_prob_2400_cal = [(y_prob_2400_sum[i] * (avg_2400/MAX_LEN)) for i in range(len(y_prob_2400_sum))]
y_prob_EA_cal= [(y_prob_EA_sum[i] * (avg_EA/MAX_LEN)) for i in range(len(y_prob_EA_sum))]
```

Figure 7: get normalization of sentence length and multiply with the sum of RoBERTa score

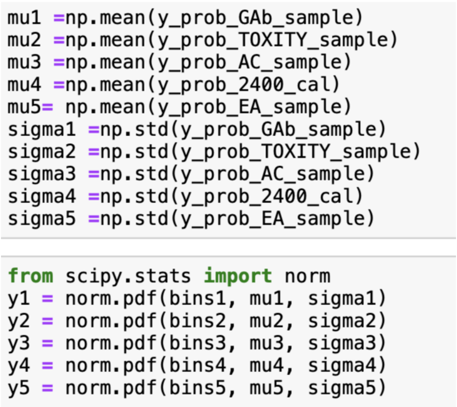
2. In order to ensure that the data volume of each database is uniform, a random sample of the same numbers as 2400 Covid data does is sampled from each database.



```
y_prob_GAb_sample = np.random.choice(y_prob_GAb_cal, size=len(y_prob_2400_sum), replace=False)
y_prob_TOXITY_sample = np.random.choice(y_prob_TOXITY_cal, size=len(y_prob_2400_sum), replace=False)
y_prob_AC_sample = np.random.choice(y_prob_AC_cal, size=len(y_prob_2400_sum), replace=False)
y_prob_EA_sample = np.random.choice(y_prob_EA_cal, size=len(y_prob_2400_sum), replace=False)
```

Figure 8: randomly get 2400 data in each data set

3. Calculate the **average** and **standard deviation** for each database, and draw the **Gaussian distribution**.



```
mu1 =np.mean(y_prob_GAb_sample)
mu2 =np.mean(y_prob_TOXITY_sample)
mu3 =np.mean(y_prob_AC_sample)
mu4 =np.mean(y_prob_2400_cal)
mu5= np.mean(y_prob_EA_sample)
sigma1 =np.std(y_prob_GAb_sample)
sigma2 =np.std(y_prob_TOXITY_sample)
sigma3 =np.std(y_prob_AC_sample)
sigma4 =np.std(y_prob_2400_cal)
sigma5 =np.std(y_prob_EA_sample)
```

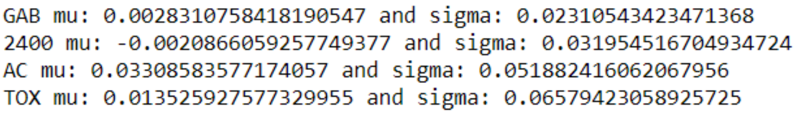
```
from scipy.stats import norm
y1 = norm.pdf(bins1, mu1, sigma1)
y2 = norm.pdf(bins2, mu2, sigma2)
y3 = norm.pdf(bins3, mu3, sigma3)
y4 = norm.pdf(bins4, mu4, sigma4)
y5 = norm.pdf(bins5, mu5, sigma5)
```

Figure 9: calculate the mean and standard deviation for each data set to plot the Gaussian distribution

Note: in the formula of Gaussian Distribution: $f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ σ indicates standard deviation of each data set, μ indicates the mean value of the sum of RoBERTa score of each data set and x indicates the bins value for each data set.

3.2 Summary

Through the observation and analysis of the above images, we can find that among the distribution curves of different databases, the Gaussian distribution curve of the Gab data has a **smaller standard deviation** and a **higher peak value**. Therefore, after this experiment, we can be preliminary concluded that after fitting the Gaussian curve to each database, a database with a flat data distribution (larger standard deviation and smaller peak value) could achieve better performance.



```
GAB mu: 0.0028310758418190547 and sigma: 0.02310543423471368
2400 mu: -0.0020866059257749377 and sigma: 0.031954516704934724
AC mu: 0.03308583577174057 and sigma: 0.051882416062067956
TOX mu: 0.013525927577329955 and sigma: 0.06579423058925725
```

Figure 10: the mean value and standard deviation value for each data set

4 Distribution analysis implemented by Auto-Encoder combined with transformer

Based on our last week discussion, there was no obvious distribution difference in the distance comparison graphs.(especially after we removing the fine-tuning operation on the linear layers).
In next step, we would construct a model by combining **transformer** and **Auto-Encoder** and follow the steps below:

1. simulate new data by extracting input data features

2. perform KL-divergence or MSE score to evaluate the loss between the data before and after the input, so as to realize whether the database features are obvious or not.

Through this operation, we respectively generate and compare the databases: Toxicity, Attack Comment, Gab, and EA data, then find a reasonable threshold value, compare the distribution of the database to be tested with this value, and judge whether the database is good or bad. (We could analogize this operation to Outlier Detection. Under normal circumstances, outlier data occupies only a small part of the total data. In our experiment, the positive data is small in scale. Therefore, the abnormal data analyzed through the model is positive data.)

4.1 Structure of Auto-Encoder

An Auto-Encoder is a neural network that learns to copy its input to its output. It has an internal (hidden) layer that describes a code used to represent the input, and it is constituted by two main parts: an **encoder** that maps the input into the code, and a **decoder** that maps the code to a reconstruction of the original input.

Performing the copying task perfectly would simply duplicate the signal, and this is why Auto-Encoder usually are restricted in ways that force them to reconstruct the input approximately, preserving only the most relevant aspects of the data in the copy.

The most traditional application of it was **dimensionality reduction** or **feature learning,** but more recently the Auto-Encoder concept has become more widely used for learning generative models of data. The schema of basic Auto-Encoder shown in the below.

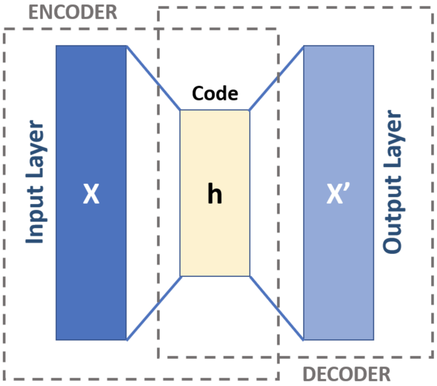
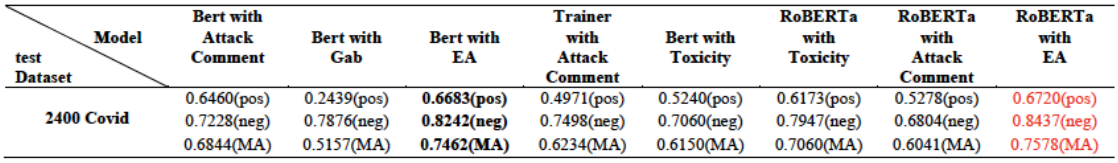


Figure 11: Schema of a basic Autoencoder

5 Appendix



Model / test Dataset	Bert with Attack Comment	Bert with Gab	Bert with EA	Trainer with Attack Comment	Bert with Toxicity	RoBERTa with Toxicity	RoBERTa with Attack Comment	RoBERTa with EA

2400 Covid	0.6460(pos)
	0.7228(neg)
	0.6844(MA)
		0.2439(pos)
		0.7876(neg)
		0.5157(MA)
			0.6683(pos)
			0.8242(neg)
			0.7462(MA)
				0.4971(pos)
				0.7498(neg)
				0.6234(MA)
					0.5240(pos)
					0.7060(neg)
					0.6150(MA)
						0.6173(pos)
						0.7947(neg)
						0.7060(MA)
							0.5278(pos)
							0.6804(neg)
							0.6041(MA)
								0.6720(pos)
								0.8437(neg)
								0.7578(MA)

Figure 12: Comparison of the results different models test on 2400 Covid data

By comparing the test results of many models on the 2400, we can find that in a new round of experiments, the RoBERTa model constructed with EA data has achieved the best results so far, which could reach 0.6720 for positive class and 0.8437 for negative class.