Work summary for the 5^{th} week

Student: Xuanyu Su
Supervisor: Isar Nejadgholi

October 5, 2020

1 Test the 'Attack Comments Model' with manually labeled 'COVID-HATE' data set(Version 2.0)

1.1 Recall of previous version

In the 4^{th} week 's work, we put **2,400** hand labeled data of 'Anti-Asian' in our previous 'Attack Comment Classification' model for testing and result evaluation. In our previous work, we re-label only the 'Hate' data into positive class '1', all the other types of data(**Neutral**, **Counterhate** and **Non-Asian agressives**) into negative '0', we were trying to improve the performance by changing the maximum length of text as well as removing the emojis, and the best result as shown in figure 1 (no emojis were removed and the max text length is 60):

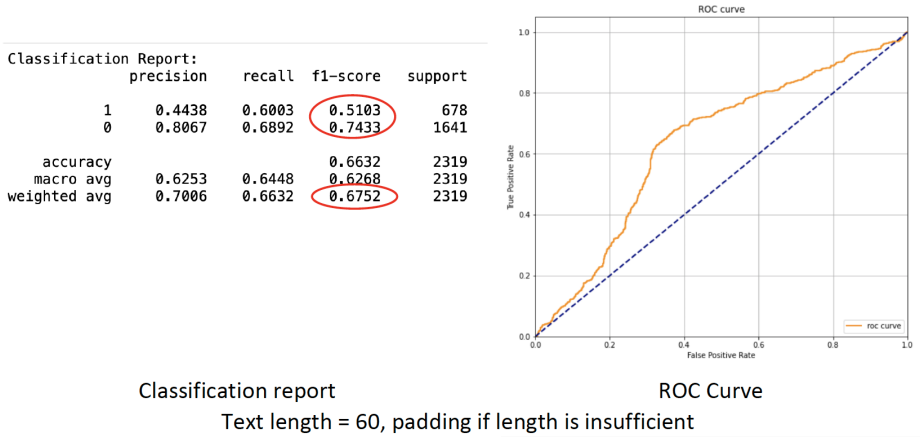


Figure 1: The classification report and ROC curve not remove emoji and text length = 60

1.2 'Re-labeled' version

In this module, we re-labeled the data from the previous 'Hate' as '1' VS 'Neutral, Conterhate and Non-Asian' as '0' to 'Hate and Non-Asian ' as '1' VS 'Neutral and Conterhate ' as '0'. The reason why we made this modification is because both the 'Hate' and 'Non-Asian' contains hate vocabularies. In the original version, we separated them into two categories, which will increase the interference to the results, so the result of the experiment is not very good. The results of our revised version are as figure 2:

Through the observation of F1 score, we can find that after the model has reclassified the labels, the F1 score of the positive class has increased from **0.5183** to **0.6468**, and the weighted average score has increased from **0.6752** to **0.6897**, although the F1 score of predicting negative classes has dropped slightly from **0.7433** to **0.7228**, overall it has been greatly improved. It shows that it is a correct and reasonable operation to re-divide the data label.

2 Trainer()

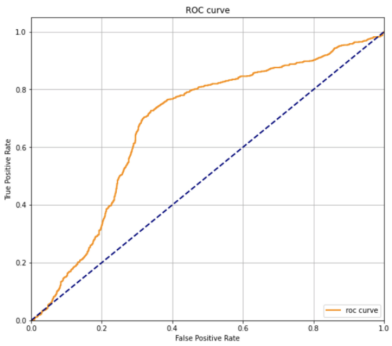
2.1 Test trainer on 'Attack Comment' data

In this module, we mainly used '**trainer()**' method to replace the 'model construction' part of our previous model. By calling the '**distilbert-base-uncased**', the amount of code was simplified. Then apply the method on the 'Attack Comment' data, the result as shown in figure 3:

Classification Report:

	precision	recall	f1-score	support
1	0.6339	0.6587	0.6460	999
0	0.7338	0.7121	0.7228	1320
accuracy			0.6891	2319
macro avg	0.6839	0.6854	0.6844	2319
weighted avg	0.6908	0.6891	0.6897	2319

The Classification report of version2



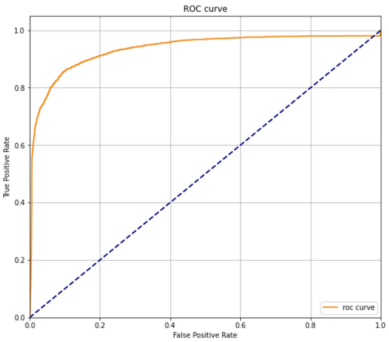
The ROC curve of version2

Figure 2: The classification report and ROC curve after modify the data label

Classification Report:

	precision	recall	f1-score	support
1	0.5455	0.0109	0.0213	2756
0	0.8821	0.9988	0.9368	20422
accuracy			0.8813	23178
macro avg	0.7138	0.5048	0.4791	23178
weighted avg	0.8421	0.8813	0.8280	23178

The Classification report of trainer on 'Attack Comment'



The ROC curve of trainer on 'Attack Comment'

Figure 3: The classification report and ROC curve by using trainer and test on 'Attack Comment' data

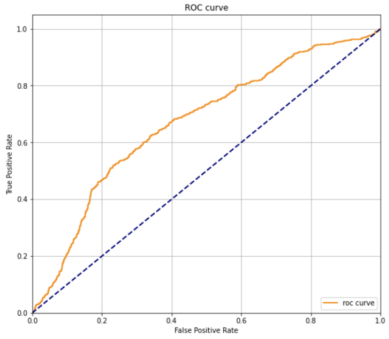
2.2 Test trainer on 2400 hand labeled 'Anti-Asian' data

In this module, we apply the '**trainer()**' method on the 'Anti-Asian' data, the result as shown in figure 4:

Classification Report:

	precision	recall	f1-score	support
1	0.2222	0.0088	0.0170	678
0	0.7068	0.9872	0.8238	1641
accuracy			0.7012	2319
macro avg	0.4645	0.4980	0.4204	2319
weighted avg	0.5651	0.7012	0.5879	2319

The Classification report of trainer on 2400 hand labeled anti-asian



The ROC curve of trainer on 2400 hand labeled anti-asian

Figure 4: The classification report and ROC curve by using trainer and test on 2400 hand labeled 'Anti-Asian' data

By comparing the above two results, we can further confirm the conclusion of what we got last week. In this module, we use trainer() method to test two different data sets and find that there is a better effect on the 'Attack comment' data set. , The F1 score of the positive class prediction is **0.8213**, but when tested with 2400 'Anti-Asian' data, the F1 score of the positive class is only **0.3170**. Hence, we can conclude that under the same model, the accuracy of the data set itself will effect the results of the experiment(that means our model was trained through the ' Attack Comment' data, so it gets better results on that data set. While, for the 'Anti-Asian' data set, since it has more detailed categories. And our original binary model cannot meet higher accuracy classification). Therefore, in the follow-up work, we hope that through the establishment of a sufficiently comprehensive and detailed database, and the continuous improvement and simplification of the model, the experimental results could be optimized.

3 Twitter API

3.1 From Tweets ID to content

In this module, we have obtained the **Twitter API** usage permission, which can help us convert the **Tweet ID** in the original database into the corresponding content, but because the **Twitter API** permission and version we currently obtain are low, we cannot convert all the data at a time.Therefore, only **614** pieces of 'Hate' data were converted. The results before and after the conversion are as follows (after the conversion, we removed the **Tweet ID** and **User ID**, as well as the probability columns, and simply retained the text content and the label).

	Tweet ID	User ID	Hate Probability	Counterhate Probability	Neutral Probability	Label
0	1242513753733808128	1037398540698312704	0.625769	0.210356	0.063975	Hate
1	1242355882119155713	1242352444136247298	0.793170	0.133605	0.008983	Hate
2	1242280806711611392	14369596	0.664656	0.447038	0.075040	Hate
3	1242484533875707910	1067189627721850880	0.798824	0.015304	0.112250	Hate
4	1242433526407917568	3139806542	0.969103	0.001253	0.092968	Hate

Figure 5: The content before using Twitter API

	Comment	attack
0	Until the world finds a cure for Covid19 .\n\n...	hate
1	@JeffreyGuterman @realDonaldTrump But it is th...	hate
2	@cpimspeak The Chinese Communist Party acted i...	hate
3	@Sar_Be_Dar Why is the US to blame that is tot...	hate
4	@MFA_China China is mother of #WuhanVirus. \n\...	hate

Figure 6: The content after using Twitter API

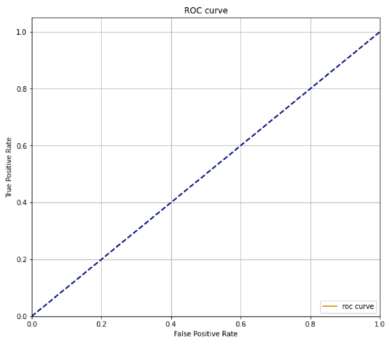
3.2 GHC data test

In this module, we test the 614 marked 'Hate' data on the previous 'Attack Comment' model. Since all of our current data has already been labeled with positive class '1'. The purpose of this test is to more fully and comprehensively analyze the performance of the model. The higher the accuracy of the model's positive class detection, the higher the accuracy of the model, and vice versa. The result as shown in figure7:

Classification Report:

	precision	recall	f1-score	support
0	0.0000	0.0000	0.0000	0
1	1.0000	0.1726	0.2944	614
accuracy			0.1726	614
macro avg	0.5000	0.0863	0.1472	614
weighted avg	1.0000	0.1726	0.2944	614

The Classification report of GHC data in 'Attack Comment' model



The ROC curve of GHC data in 'Attack Comment' model

Figure 7: The classification report and ROC curve after test the transferred data on 'Attack Comment' model

As can be seen from the results in the above figure, the model's prediction for negative classes is 0, and the prediction for positive classes is 0.2988, indicating that our current model only has 0.2988 accuracy to detect these hate words. This also tells us that our model needs to be trained with better and more accurate data to get better results.