Work summary for the 7th and 8th week

Student: Xuanyu Su
Supervisor: Isar Nejadgholi

October 26, 2020

1 Bert model construction with 'Toxicity' data

1.1 Data Preprocessing

1. **Voting the duplicate texts**: since in this database, the data is manually labelled. Therefore, there is a problem with data duplication. Here we continue to use the '**voting**' method of the same as the previous model to average the label of the same data and set the threshold to **0.5**. When the average is greater than **0.5**, all the duplicate text will be relabeled as positive 'Toxicity', then after we remove duplicates text.
2. **Remove URL**: remove all the URL links;
3. **Remove punctuation**: remove all the punctuation, like: ',' , '.', '!', etc.
4. **Remove usename**: remove the usernames that have been mentioned in the texts;
5. **Remove Hashtag**: remove the hashtags mentioned in the texts;
6. **Transform uppercase to lowercase**: transform all the uppercase into lowercase;
7. **Remove special words**: 'NEWLINE TOKEN;
8. **Merge tables**: merge the 'toxicity-annotated-comments' table with 'toxicity-annotations' table based on 'rev-id'.
9. **Split train, valid and test data sets**: split data set into train, validation and test sets by keyword query in the 'split' column.
10. **Drop unrelated columns**: after we finish all the above operations, we remove all the unrelated columns: 'rev-id' ,'year', 'logged-in', 'ns', 'sample', 'worker-id', and 'toxicity-score'.

1.2 Model used and result

In this version of the model construction, we still use the same model framework as the previous '**Attack Comment**' model: use the **fields** and **TabularDataset** methods to do data embedding process, and then imported data into the **bert-base-uncased** model, set the criterion to **BCELoss()**, the optimizer to **AdamW**, and **weight-decay** is used to prevent overfitting. The loss curve of the model is as figure 1:

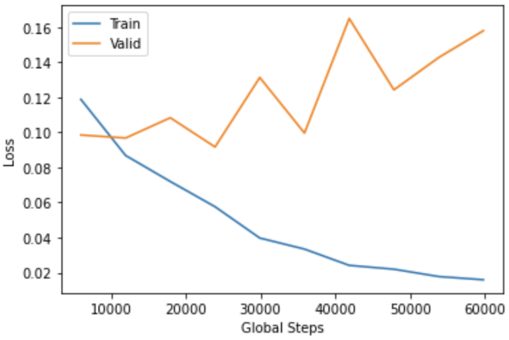
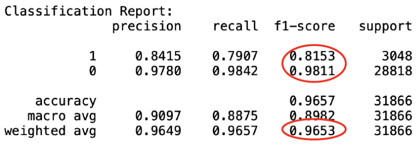


Figure 1: The loss curve of model built with toxicity data

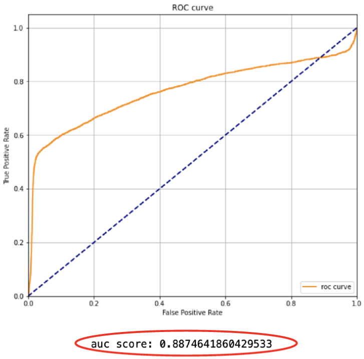
The results of in-domain and testing on 'Gab', 'Attack Comment' and '2400 Covid' data as shown in the below:



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

           1     0.8415    0.7907    0.8153      3048
           0     0.9780    0.9842    0.9811     28818

    accuracy                         0.9657     31866
   macro avg     0.9097    0.8875    0.8982     31866
weighted avg     0.9649    0.9657    0.9653     31866
```

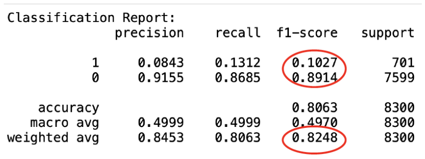


```
auc score: 0.8874641860429533
```

The classification report of model **in-domain** test

The roc curve of model **in-domain** test

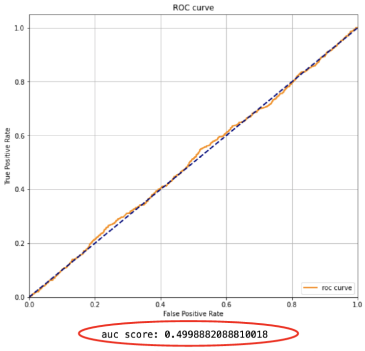
Figure 2: The classification report and ROC curve of model in-domain test



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

           1     0.0843    0.1312    0.1027       701
           0     0.9155    0.8685    0.8914      7599

    accuracy                         0.8063      8300
   macro avg     0.4999    0.4999    0.4970      8300
weighted avg     0.8453    0.8063    0.8248      8300
```

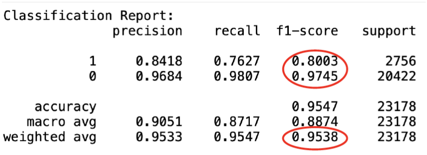


```
auc score: 0.4998882088810018
```

The classification report of model test on **Gab** data

The roc curve of model test on **Gab** data

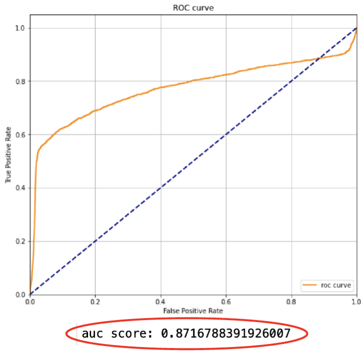
Figure 3: The classification report and ROC curve of model test on 'Gab' data



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

           1     0.8418    0.7627    0.8003      2756
           0     0.9684    0.9807    0.9745     20422

    accuracy                         0.9547     23178
   macro avg     0.9051    0.8717    0.8874     23178
weighted avg     0.9533    0.9547    0.9538     23178
```

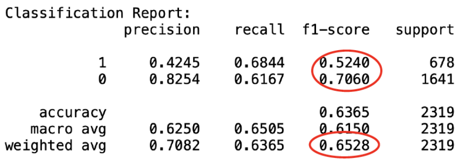


```
auc score: 0.8716788391926007
```

The classification report of model test on **Attack Comment** data

The Roc curve of model test on **Attack Comment** data

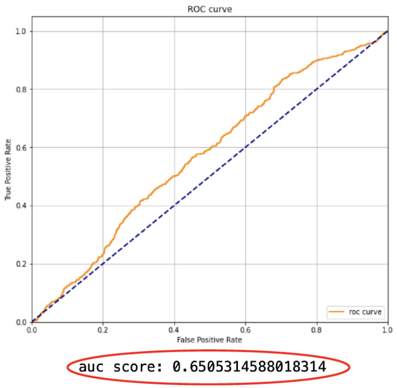
Figure 4: The classification report and ROC curve of model test on 'Attack Comment' data



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

           1     0.4245    0.6844    0.5240       678
           0     0.8254    0.6167    0.7060      1641

    accuracy                         0.6365      2319
   macro avg     0.6250    0.6505    0.6150      2319
weighted avg     0.7082    0.6365    0.6528      2319
```

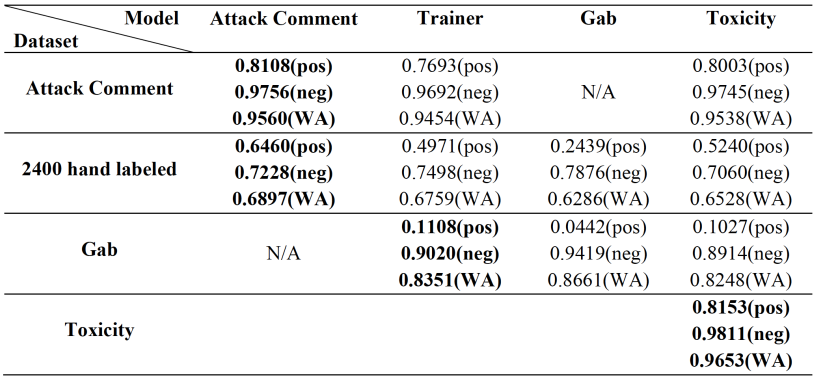


```
auc score: 0.6505314588018314
```

The classification report of model test on **2400 hand labelled** data

The Roc curve of model test on **2400 hand labelled** data

Figure 5: The classification report and ROC curve of model test on '2400 hand labelled' data



Model Dataset	Attack Comment	Trainer	Gab	Toxicity

Attack Comment	**0.8108(pos)**
	0.9756(neg)
	0.9560(WA)
		0.7693(pos)
		0.9692(neg)
		0.9454(WA)
			N/A	0.8003(pos)
				0.9745(neg)
				0.9538(WA)

2400 hand labeled	**0.6460(pos)**
	0.7228(neg)
	0.6897(WA)
		0.4971(pos)
		0.7498(neg)
		0.6759(WA)
			0.2439(pos)
			0.7876(neg)
			0.6286(WA)
				0.5240(pos)
				0.7060(neg)
				0.6528(WA)

Gab	N/A	**0.1108(pos)**
		0.9020(neg)
		0.8351(WA)
			0.0442(pos)
			0.9419(neg)
			0.8661(WA)
				0.1027(pos)
				0.8914(neg)
				0.8248(WA)

Toxicity				**0.8153(pos)**
				0.9811(neg)
				0.9653(WA)

Figure 6: The comparison of four model test on four data sets

1.3 Summary

Through the comparison table above, we can see that toxicity performance well on the four databases, except for the 'Gab' data. Therefore, after the above comparison, it is found that the 'Gab' database does not have high recognizability, and it cannot help construct a model with high accuracy either. Hence, in the next section, we expect to construct a larger database by extracting racism words related data from both 'Toxicity' and 'Gab' data sets. Observe and compare the performance of the new model on different databases.