Attack comments classification by using BERT

Student: Xuanyu Su
Supervisor: Isar Nejadgholi

September 14, 2020

1 Intoduction

In this assignment, we are given three datasets: $attack-annotations$, $attack-annotated-comment$ and $attack-worker-demographics$. We will use the three csv tables to do attack language detection by using BERT model with the latest version of transformer and tokenizer. We use an AUC score and confusion matrix to test whether the model is good enough. Details shown in the below.

2 Data Preprocessing

1. Merge tables and remove duplicates: In this section, we mainly focus on the annotation and comments table, since we only do the binary classification. After observing the tables, we merge the 'attack-annotations' and 'attack-annotated-comment' tables on 'rev-id' and then remove the duplicates in the tables.

There are some special words like, 'NEWLINE-TOKEN' and 'TAB-TOKEN'. In order to reduce the interference of these special characters on the experimental results, we replaced these characters with spaces(' ').

2. Data split: In this section, we split the merged table into three datasets: train, validation and test based on the column 'split', since there are three labels in the column. Then remove the original 'split' column. Check the class distribution to get the maximum length of sentences.

3 Build BERT model and set parameters

We use the default $BERT_{base}$ model and 'BertForSequenceClassification'(Bert+Pooled output from [CLS]+ dropout + dense + Sigmoid(activation functuin) Loss: Crossentropy) as the encoder. We define a checkpoint function to prevent data loss caused by operation interruption.

4 Train and Evaluate model

We imply the model on the train set firstly and then evaluate the result for every epoch, then do this on the validation set and evaluate the results as well.

In this experiment, since out data has imbalanced True and False classes, so it is better to use F1 score or AUC score to do evaluations rather than traditional accuracy. The final result shows in the below:

4.1 F1 score

Classification Report:

	precision	recall	f1-score	support
1	0.8150	0.4179	0.5525	3869
0	0.8936	0.9810	0.9352	19275
accuracy			0.8868	23144
macro avg	0.8543	0.6994	0.7439	23144
weighted avg	0.8804	0.8868	0.8713	23144

Figure 1: The classification results

4.2 Loss curve

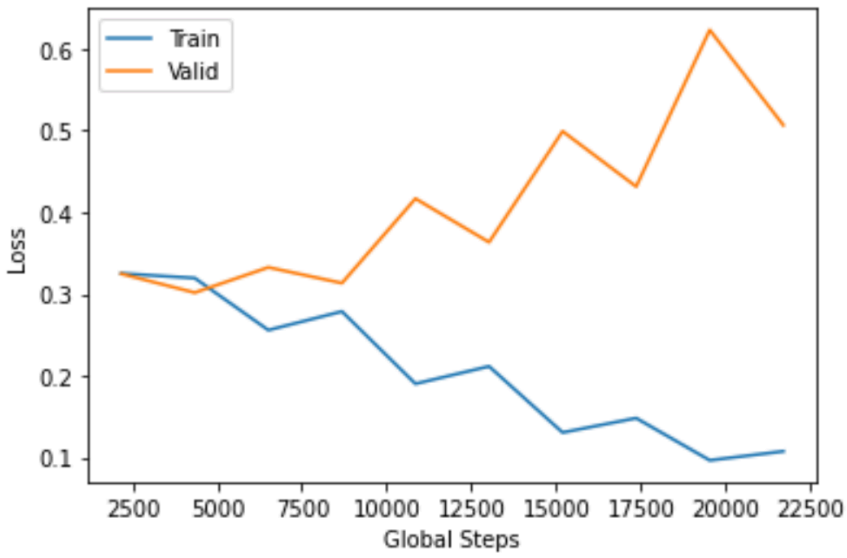

Figure 2: The loss curve for train and validation dataset

4.3 ROC curve and AUC score

With the AUC score: 0.6994.

5 Summary

Through the observation and analysis of the experimental results, especially the display of the loss curve, the model fits well on the training data, and as the model training loss gradually

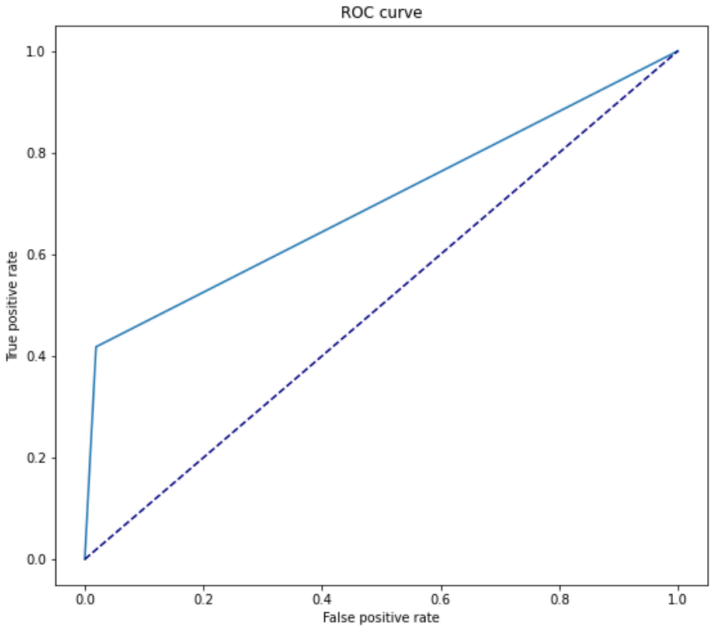

Figure 3: The ROC curve of the results

decreases, the model performs poorly on the verification data. In this case, the model should have overfitting. In the experiment, we directly call the 'BertForSequenceClassification' method, which already contains a dropout layer, so we ruled out the possibility of improving the result by adding a drop layer. This problem may be caused by the following reasons

1. The class distribution of the original dataset is imbalenced. From the first part, we can know that the proportion of attack comments account for about 16% of the total, and the non-attack comments are 84%, and in an ideal state, the ratio of positive and negative data should be 1:1. The insufficient training data causes the model to be unable to get better results.

2. Failure to clean up the interference information in the data preprocessing stage. In our version of the experiment, due to time constraints and the amount of data being too large, we only initially delete duplicate data and deal with part of special characters. More operations require careful reading of the data in the future improvement.

6 Possible future work.

1. The data from the above results shows that our model does not perform well in 'recall score' (that means the model can not identify all the comments that marked with 'attack'). By referring to the worker table, it is not difficult to find that different users have different educational backgrounds, which arouses our thinking: language attacks can be divided into implicit and explicit, our model can only identify the latter at present, that is, vocabulary that is obviously offensive and unpleasant. Many offensive words with implicit or suggestive meanings are ignored, so this has become one of the factors we need to consider in the future.

2. In this experiment we simply do a binary classification, but there are detailed attack type in the table, so maybe in the next, we will build a multi-class model to identity which specific

type the attack comments is. And this will also have more benefit on social especially the network security as well.