Work summary for the 4^{th} week

Student: Xuanyu Su
Supervisor: Isar Nejadgholi

September 28, 2020

1 Test the 'Attack Comments Model' with manually labeled 'COVID-HATE' data set

In this module, we put **2,400** test data of Anti-Asian in our previous 'Attack Comment Classification' model for testing and result evaluation. Before drawing conclusions, we performed the following preprocessing on the test data.

1.1 Data Preprocessing

1. First, we observed the data and found that because the data mainly comes from **Twitter**, the data contains many emoticons such as **emoji**. We consider that people will convey emotions and opinions through **emojis**, Therefore, we convert these emojis into corresponding words through calling related packages.
2. We removed all Twitter- and web-specific content such as **URLs**, **usernames**, **hashtags**.
3. We also removed irrelevant **punctuation** and converted all **uppercase** letters to **lowercase** letters.
4. Our test data is a **multi-class label**, and the model is a **binary model**. Due to the limitations of current knowledge, we can only choose the most important class from the four labels of test data as the positive class(in this experiment, we use the class that labeled 'Hate' as the positive class, labeled with '1'), and convert the other classes to negative, labeled with '0', then put the data into the model for testing.

1.2 Model training and results

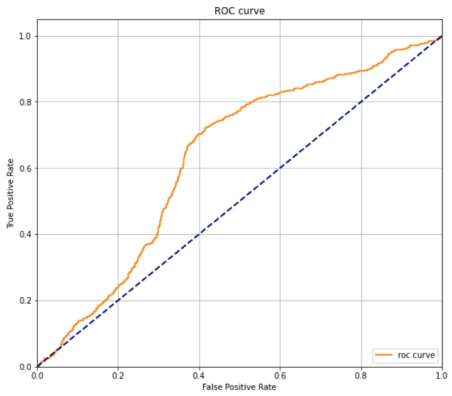
1.2.1 No emojis removed

It should be pointed out that in this module, we conducted three versions of testing and comparison, all based on the premise that emoji was not removed.

1. After preprocessing the data in the previous step, we got the first result when not remove emoji and the text length is 128(padding if the length is insufficient), details shown in Figure 1.
2. After the above test, we found that the result of the experiment is not very good. Then we have analyzed the length distribution of the text and found that the maximum text length of the original model is 128, but the maximum length of our test data is 60, we made the following assumptions: when the maximum length is too large, padding will increase the interference to the result. Therefore, in the second version of the test, we adjusted the maximum length of the text to 60, and the results are as Figure 2.
3. By comparing the results of version 1 and 2, we found that the weighted accuracy of the model and the F1 score of the negative class (data labeled with '0') have improved slightly after the maximum length of the text is modified into 60, but the F1 score of the prediction of the positive class(date labeled with '1') has decreased. Therefore, while keeping the original model unchanged(keep the maximum length equal to 128 of the original model), we adjust the length of test data: since the length of our test data is around 60, we doubled the length by copying the text once, then the length of each text is close to 128, by copying the text with insufficient length to reduce the interference of padding. And the results are as Figure 3.

Classification Report:

	precision	recall	f1-score	support
1	0.4330	0.6578	0.5222	678
0	0.8200	0.6441	0.7215	1641
accuracy			0.6481	2319
macro avg	0.6265	0.6510	0.6219	2319
weighted avg	0.7069	0.6481	0.6632	2319



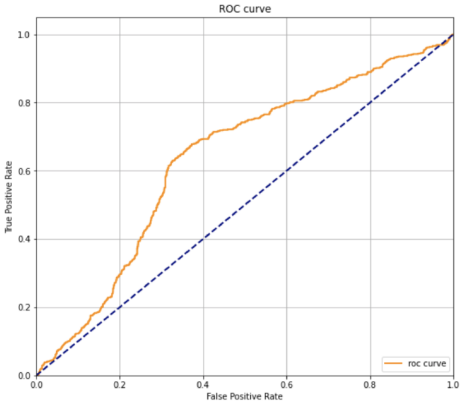
Classification report ROC Curve

Text length=128, padding if length is insufficient

Figure 1: The classification report and ROC curve not remove emoji and text length = 128

Classification Report:

	precision	recall	f1-score	support
1	0.4438	0.6003	0.5103	678
0	0.8067	0.6892	0.7433	1641
accuracy			0.6632	2319
macro avg	0.6253	0.6448	0.6268	2319
weighted avg	0.7006	0.6632	0.6752	2319



Classification report ROC Curve

Text length = 60, padding if length is insufficient

Figure 2: The classification report and ROC curve not remove emoji and text length = 60

1.2.2 Remove emojis

Through the result of the third version, we found that the results of the experiment were not improved, but lower than the results of both the previous two. Therefore, we concluded that padding does not interfere with the model test results.

After reading the paper detailed, it was discovered that the author **removed emoji** when processing data, so we performed the same operations. After removing the emoji and not changing the maximum text length, the experimental results are shown in the figure 4 below.

1.3 Summary and conclusions

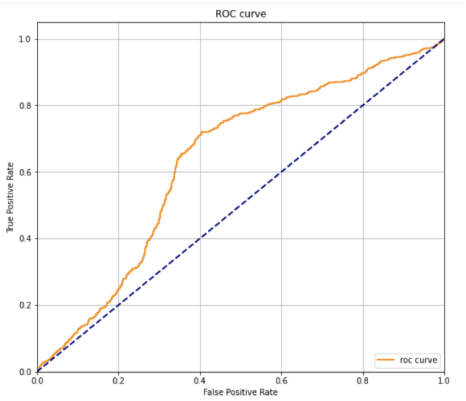
Through our continuous improvement and exploration of the model, we found that the model did not show significant improvement and optimization for the above operations.

Combined with the characteristics of the test data itself, we have reached a preliminary conclusion: our original 'Attack Comments Classification' model is suitable for predicting **binary** data, and our now data, although processed, due to the complexity of the data itself (the database contains four categories, of which **Hate** and **Non-Asian agressive** both contain hate vocabularies, but when we redefine the label, the **Hate** is marked as a positive class '1'. Others such as **Neutral**, **Counterhate** and **Non-Asian agressives** are all marked as negative class '0'), our model failed to achieve good experimental results.

Therefore, we infer that the model needs to be trained on more accurately classified data in order to achieve higher generalization capabilities and achieve the purpose of prediction and evaluation of different types of text. In other words, our current work is: trying to use a model trained with binary data to test multi-labeled (more complex) data,

Classification Report:

	precision	recall	f1-score	support
1	0.4269	0.6504	0.5155	678
0	0.8157	0.6392	0.7168	1641
accuracy			0.6425	2319
macro avg	0.6213	0.6448	0.6161	2319
weighted avg	0.7020	0.6425	0.6579	2319



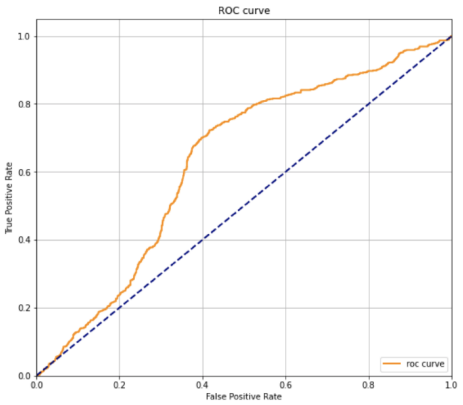
Classification report ROC Curve

Text length = 128, copy the text if length is insufficient

Figure 3: The classification report and ROC curve not remove emoji and copy text to length = 128

Classification Report:

	precision	recall	f1-score	support
1	0.4326	0.6622	0.5233	678
0	0.8212	0.6411	0.7201	1641
accuracy			0.6473	2319
macro avg	0.6269	0.6517	0.6217	2319
weighted avg	0.7076	0.6473	0.6625	2319



Classification report ROC Curve

Text length = 128, remove emojis

Figure 4: The classification report and ROC curve after removing emojis

so the model is difficult to achieve the expected result. However, through the above exploration and conclusions, we can train the model with multi-label (more specific and detailed classification) data in the future to make the model more sensitive and accurate to achieve better results.

2 Modify 'Attack Comments' model with Trainer()

2.1 Usage and results

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. After we tested the data with trainer(), the results are as Figure 5 and 6:

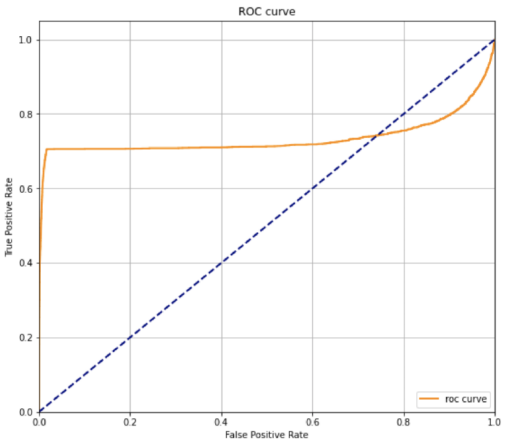
2.2 Problems encountered

From Figure 5 and Figure 6, we can find that the accuracy of the model in predicting the positive class is only **0.19**, which shows that the trainer() method did not fit well in the given data. At the same time, in our initial model construction and data import process, we encountered the problem of data type mismatch and unable to predict (we were trying to fix it, but it takes time and we finally over come it by modifying the steps of converting the input data type in the trainer method).
Considering the low accuracy of the model on 'IMDb' data and the limitation of time factors, we haven't test the 2400 data.

Classification Report:

	precision	recall	f1-score	support
1	0.2761	0.1557	0.1991	2756
0	0.8924	0.9449	0.9179	20422
accuracy			0.8511	23178
macro avg	0.5842	0.5503	0.5585	23178
weighted avg	0.8191	0.8511	0.8324	23178

Classification report of Trainer()



ROC curve of Trainer()

Figure 5: The classification report and ROC curve after applying trainer()

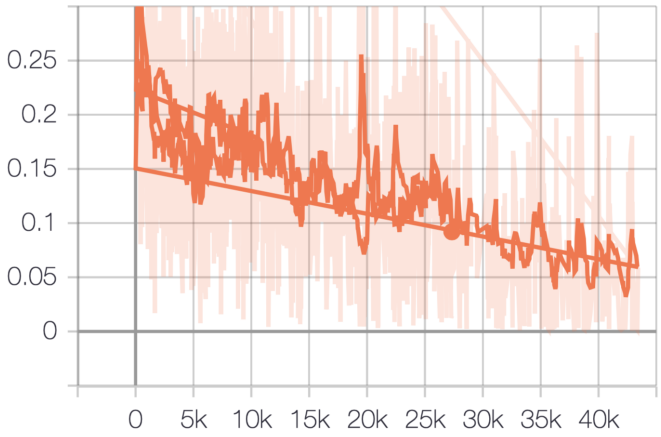


Figure 6: The loss curve after removing emojis

3 AutoML method

3.1 Introduction

In a traditional machine learning application, practitioners have a set of input data points to train on. The raw data may not be in a form that all algorithms can be applied to it. To make the data amenable for machine learning, an expert may have to apply appropriate **data pre-processing**, **parameter engineering**, **parameter extraction**, and **parameter selection** methods. After these steps, practitioners must then perform algorithm selection and hyperparameter optimization to maximize the predictive performance of their model. All of these steps induce challenges, accumulating to a significant hurdle to get started with machine learning.

3.2 The Principles of AutoML Implementation

1. **Neural Architecture Search** (NAS), it realizes the automatic generation of neural network structure through a certain structure and algorithm. It mainly includes search space, search strategy, performance evaluation strategy and other dimensions of knowledge.

2. **Hyperparameter optimization**: automatically optimize the hyperparameters in the neural network. Through the algorithm level, the hyperparameter settings are handed over to the computer to search and obtain. The more popular algorithms mainly include: **Bayesian optimization**, **random search**, and **grid search**.

3. **Meta-Learning**: is described as the process of learning from previous experience gained during applying various learning algorithms on different types of data, and hence reducing the needed time to learn new tasks.

3.3 AutoML platform

1. Auto-sklearn;
2. H2O AutoML;
3. AutoWEKA;
4. Google Cloud AutoML.

Due to the limitation of conditions, and most platforms are charged. We currently only gather the basic knowledge of AutoML. If necessary in the future, we will choose the platform to build the model according to the situation.

4 Summary of 'The Gab Hate Corpus' Project

4.1 Summary of paper

The Gab Hate Corpus (GHC), consisting of **27,665** posts from the social network service **gab.ai**, each annotated by a minimum of three trained annotators. Annotators were trained to label posts according to a coding typology derived from a synthesis of hate speech definitions across legal, computational, psychological, and sociological research. The GHC, which is the largest theoretically-justified, annotated corpus of hate speech to date, provides opportunities for training and evaluating hate speech classifiers and for scientific inquiries into the linguistic and network components of hate speech. The paper can be summarised into the following steps:

1. review and synthesize prior work in multiple disciplines to develop a theoretically-justified typology and coding guide.
    - Hate-based rhetoric: A document can be (1) Not-hateful, (2) Incitement to hatred/Call to Violence; and/or (3) Assault on Human Dignity.
    - Vulgarity/Offensive Language: A document can use offensive or abusive language which may or may not be one of the above hate-based categories
    - Targeted Group: The type of targeted group
    - Implicit/Explicit: Whether the rhetoric is direct and explicit, or it is veiled and reliant on external information to accomplish its objective.

    The full typology and workflow that annotators were to follow is visualized in Figure 7 :

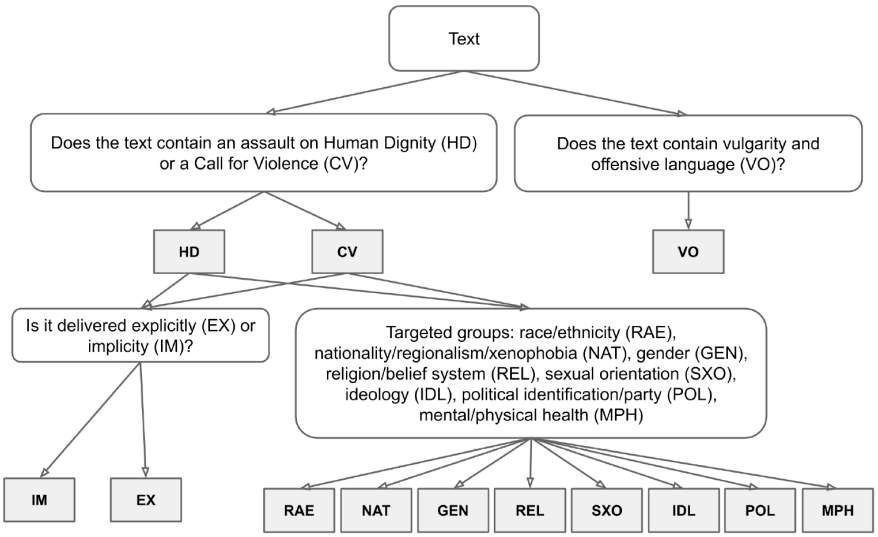


Figure 7: The typology workflow

2. rigorously train a cohort of undergraduate research assistants to accurately identify hate-based rhetoric based on the developed coding guide.

3. annotate 27,665 posts from the social network platform Gab by a minimum of three trained annotators per post.

4. run baseline as well as state-of-the-art machine learning models to classify the entirety of the Gab corpus.
   In order to establish classification baselines, here the author provide **cross-validated metrics** of performance for three methods which are representative of many of the standard approaches used in NLP:

   - bag of words modeling,
   - dictionary-based measures,
   - language model fine-tuning.

   They apply these methods to the HD and VO labels, and they also train models on the "Hate" label, which is the annotator-level union of HD and CV, given that CV labels are too sparse to provide enough signal to train predictive models.
   After evaluating each classifier, then apply it to the full dataset of Gab posts, in order to estimate the distribution of hate-based rhetoric on the entire social network. Using retrained models using the best-fit parameters of the **TF–IDF** and **LIWC** models, and the best-performing fine-tuned **BERT** model weights (by F1), result shown in Figure 8.

Model	Label	F_1	Accuracy	Precision	Recall
LIWC	HD	0.26 (0.01)	0.71 (0.01)	0.60 (0.03)	0.17 (0.01)
	VO	0.33 (0.02)	0.85 (0.01)	0.58 (0.03)	0.23 (0.01)
	Hate	0.39 (0.02)	0.71 (0.01)	0.56 (0.03)	0.30 (0.02)
TF-IDF	HD	0.40 (0.02)	0.84 (0.01)	**0.62 (0.03)**	0.29 (0.02)
	VO	**0.42 (0.03)**	0.89 (0.01)	**0.65 (0.05)**	0.31 (0.02)
	Hate	0.53 (0.02)	0.81 (0.01)	**0.66 (0.02)**	0.44 (0.01)
BERT	HD	**0.44 (0.03)**	**0.92 (0.01)**	0.46 (0.03)	**0.42 (0.02)**
	VO	**0.42 (0.03)**	**0.94 (0.00)**	0.45 (0.03)	**0.39 (0.03)**
	Hate	**0.58 (0.02)**	**0.87 (0.01)**	0.58 (0.02)	**0.57 (0.02)**

Figure 8: Mean and standard deviation of F1, precision, recall, and accuracy for predicting HD, VO, and Hate (union of HD and CV) across 10-fold cross validation.

4.2 Data

Through the observation of the database, it contains two csv tables: **GabHateCorpus-full-predictions** and **GabHateCorpus-annotations.**. In the **GabHateCorpus-full-predictions** table, it shows the labeling results of both **BERT** model and TF-IDF **model** on the three types of data.
we found that in the **GabHateCorpus-annotations** table except the label whether the text belongs to the **hate** or not, there are 13 other subcategories represented by abbreviations. Their specific meanings are as follows:

1. **HD**: Assaults on human dignity;
2. **CV**: Calls for violence;
3. **VO**: Vulgarity and/or Offensive language;
4. **RAE**: Race or ethnicity (includes anti-asian, anti-latino, anti-black, anti-arab, anti-semitism etc.);
5. **NAT**: Nationality/regionalism (includes general xenophobia and targets against countries/regions);
6. **GEN**: Gender (anti-woman, anti-man, anti-trans etc.);
7. **REL**: Religion/spiritual beliefs (anti-muslim, anti-christian, etc.);
8. **SXO**: Sexual Orientation;
9. **IDL**: Ideology (conservative/liberal/leftist/right-wing);
10. **POL**: Political identification. Includes any reference to membership in a political organization (Democratic/Republican/ etc.);

11. **MPH**: Mental/physical health status, physical disability;
12. **EX**: Explicitly;
13. **IM**: Implicitly.

Note: In this week's task, we came into contact with previously unknown technologies such as trainer() method and AutoML technology. At the same time, many professional vocabulary related to the field of linguistics appeared in the paper. Therefore, it took a lot of time to get through with them, which made it impossible to complete all tasks such as the realization of topic modeling and AutoMM, if possible, we hope to have a more in-depth understanding and application of these models and methods in the next time.