EmotionGIF 2020 Challenge: A RoBERTa-based Multi-label Classifier

Meng-Shiun Tsai, Pei-Ze Chiang, Yu-Hsuan Wu
National Chiao Tung University
Hsinchu, Taiwan

Abstract

In this paper, we present our idea and approach for the EmotionGIF 2020 challenge. The task of this challenge is to predict the category of a GIF response for the given unlabeled tweets. We take advantage of two different powerful pre-trained language representation models, which are BERT and RoBERTa, to make the training process easier. Using them as backbone networks, we then fine-tune the model to handle the classification task. For the dataset, we apply some augmentation and preprocessing methods to deal with the data imbalance problem. Our RoBERTa-based model finally achieves above-average performance with the MAR score of 0.5824 on the evaluation data in this challenge.

1 Introduction

Tweet emotion analysis is a common problem in the NLP field. The goal of the EmotionGIF challenge is to recommend GIF categories for the given tweet. This challenge can be treated as a multi-label classification problem. Emotion classification has been studied for a long time. With the evolution of computing ability, we can use deep learning models to further improve its performance. Recently, a lot of powerful models have been released such as BERT, ELMo, and GPT-3. [Devlin et al., 2018, Peters et al., 2018, Brown et al., 2020] All of them use numerous training data and countless GPUs to train their model. It is almost impossible to train those models from scratch by using just personal computers. Fortunately, they released the pre-trained weight for the public. Soon, those models are fine-tuned for many tasks, which accelerates the development of the NLP field.

Emotion classification is a hard task for a computer, even for humans to understand the given sentence. Emotions involve many different aspects, such as mind, body, or culture. [Bazzanella, 2004] As a result, the classification task became even harder for just using words to identify the emotions. Before Twitter became popular, there are already existing the task of analysis "Microblogging" post. [Li and Xu, 2014] Microblogging is a form of short text, letting people share about their life. Twitter now is one of the most popular social platforms. Of course, it inherits the properties of microblogging. Tweets contain any kinds of topics, informal writing style, and local culture, so preprocessing became an important step in the tweet analysis problem. We combine many kinds of data normalization methods and try to reduce the variety of data.

In the following sections, we will introduce our methods to improve the classification result in three aspects:

- Use different augmentation tools and methods to deal with the data imbalance issue.
- Design the best preprocessing procedure on tweet data and do the ablation study.
- Fine-tune different pre-trained models (BERT and RoBERTa) on the multi-label classification tasks.

2 Related Work

2.1 BERT

BERT, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers, is a pre-training language representation method released by Google researchers. BERT makes use of Transformer, an attention mechanism that learns contextual representations of words. It only uses the encoder part of Transformer and is pre-trained on a large corpus of unlabeled text data. BERT can be used for a wide variety of downstream NLP tasks. To apply BERT on different domain problems, the only thing we need to do is to add a small layer to the core model.

2.2 RoBERTa

RoBERTa stands for **R**obustly **o**ptimized **BERT** **a**pproach. The authors of this paper found that BERT was undertrained so they improved the training process of the original BERT models. The main differences between BERT and RoBERTa include:

(1) RoBERTa was trained on longer sequences with bigger batches and more data for longer training time; (2) dynamic masking instead of static masking; and (3) removing the NSP loss. These modifications help improve the performance of the downstream tasks.

3 Proposed Method

3.1 Data Imbalance

After doing a simple statistics of the number of categories in this EmotionGIF dataset, we found that there is a very severe data imbalance problem. As shown in Figure 1, the most category in the dataset is "applause", and the least one is "mic drop". Data imbalance issues will influence the model training not tend to predict categories with less data, so we apply data augmentation to handle this problem. We use the python library called "nlpaug" to simply augment the categories which don't have as much data as the most category. We randomly pick a tweet from the category we want to augment and apply the augmentation function to get a new tweet. The new tweet we will assign the same labels as the original tweet. After doing augmentation, each of the categories has at least 3442 data. Due to the multiple labels for every data, it's hard to get them perfectly balanced. The new distribution is shown in Figure 2

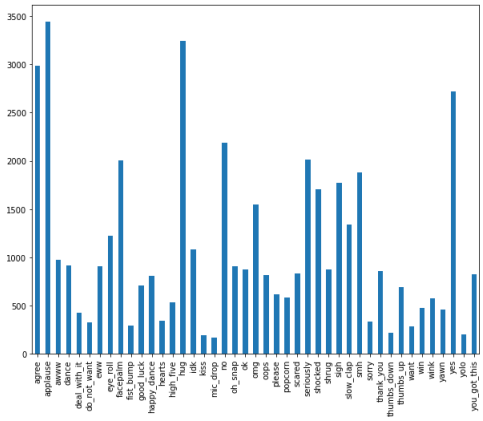

Figure 1: The raw data distribution

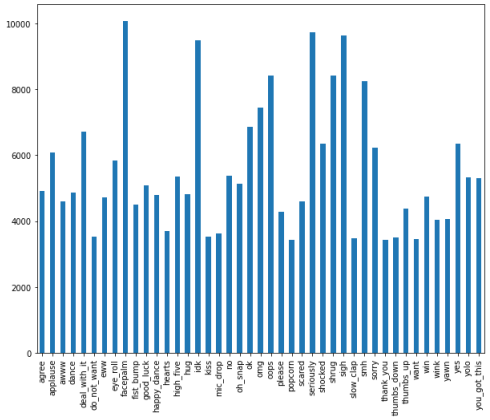

Figure 2: The distribution of augmented data

3.2 Ambiguous Data

We grab a tweet with a reply and the label as a training pair, but we found a problem that may let the model hard to train. In this competition, we are not going to use GIFs as input features. Therefore, any reply that only contains a single GIF will become an empty string. Then we will have the following situations because data is labeled under the consideration of GIF. The first case, same tweet with different reply have different labels. There are many the same content of tweet data, the most common tweet has 247 data in the dataset. We thought case one is reasonable because we can have different opinions on one topic. In the second case, the same tweet with empty or same reply has different labels. This case is caused by invisible GIF or wrong labeling. This case is the main issue we want to fix. Every time model encounters the same training data, but the ground truth is not always the same. Such a case will let the model hard to converge. We try to combine all the labels of the same tweet with the reply, and set a threshold of 30% to filter out the labels with only a few cases.

3.3 Preprocessing

Preprocessing is an indispensable step in NLP tasks. Language is very interesting and can variously express the same meaning. Preprocessing is a procedure of normalizing data to turn them into a uniform expression, then reduce the ambiguity of the dataset.

3.3.1 Quotation Mark

The different quotation marks can be caused by operating systems, personal habits, or typing errors. Sometimes it will cause searching problems, so we unify ” “ ” ‘ ’ to ’. This step will also help the

other preprocessing step, like replace abbreviation. We don't need to consider other possible cases to match the pattern we want.

3.3.2 Lowercase

In some cases, the user will use uppercase to emphasize their emotion or expression. This may not follow the classic capitalization rules. Not to mention the tweet users, sometimes they don't even follow the grammar. We don't want to create a word embedding for each variation caused by uppercase. We turn all the words in lowercase to deal with this problem.

3.3.3 Demojize

Emoji is such a useful and time-saving invention in social media, sometimes it can express more meaning than words. Humans can perceive more emotion from the face, but hardly from words. That's why we often misunderstanding others' intentions on email or chatting online. For computers, emoji just a bunch of ASCII codes. We use the function of python library "emoji" to turn emoji into words. We hope this will give us more information.

3.3.4 Abbreviation

The tweet is an informal message that you post on twitter to let your friends or fans to see. It can be in any form or anything you want to say. The abbreviation is widely used in tweets. There are too many kinds of abbreviations, and it will also change over time. We only replace those are relatively common cases in our data.

Abbrev	" u "	"lol"	"idk"	..
Text	you	laugh out loud	i don't know	..
# num	583	536	117	..

Table 1: Preprocessing - abbreviation

3.4 Model

We apply transfer learning for this task, which means that we use the pre-trained model such as BERT and RoBERTa, add an untrained layer on the end of the model, and train the new classification model. This fine-tuning process helps lower the training cost. Using this technique, we don't need to train the entire model from scratch. The huggingface PyTorch library provides several interfaces for various NLP tasks. We choose the one called BertModel, which outputs raw hidden-states without any specific head on top. Then, we simply add a linear classifier layer to make the model suited to our task. The model architecture is shown in Figure 3.

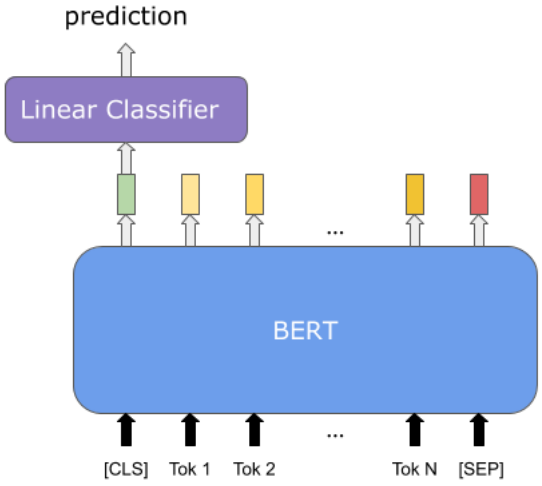

Figure 3: Model Architecture

4 Experiments

4.1 Dataset

EmotionGIF dataset contains 40,000 tweets and is separated into three parts. They are training, development, and evaluation part. Each of them has 32,000, 4,000, and 4,000 tweets respectively. We get golden labels of training data, and each data pair contains a tweet and a reply. There are 43 categories of emotions, and each sample has 1 to 6 categories as labels.

4.2 Evaluation Metrics

To evaluate the performance of our model, this competition uses mean average recall at 6 (MAR@6) score. We need to predict the six most likely labels for each sample in the evaluation dataset. Then, the recall will be computed as follows:

$$Recall = \frac{|G \cap P|}{|G|}$$

The final recall score will be calculated by averaging the recall score of each sample. The submissions will be ranked based on this score.

4.3 Experimental Settings

To train a multi-label classification model, we apply following settings:

- batch size: 8
- learning rate: 1e-5, 5e-4
- number of epochs: 6

Note that there are two learning rates, 1e-5 and 5e-4. The pre-trained part of the whole model uses

1e-5 and the classifier part uses 5e-4. The reason we apply two different learning rates is that the classifier part is just like the common classification model used in other fields. We don't have to prevent it from the catastrophic forgetting problem ([Sun et al., 2019]), which the pre-trained part may suffer from.

4.4 Results

4.4.1 Preprocessing and Using BERT-Base

To make sure that each step in our training procedure can improve the MAR score, we do the ablation study on our tweets preprocessing method and dataset preprocessing method. In the tweet preprocessing method, as shown in table 2, we find that concatenating reply to text can improve MAR score 0.59%. After that, unifying quotation marks and turning all tweets in lowercase can improve MAR score 0.55%. Because emojis are commonly used in tweets, demojizing also improve MAR score 0.44%. However, replacing abbreviations into normal words drops the score a little. Turning them into several individual words may weaken their connection. In the dataset preprocessing method, as shown in table 2, if we only consider the situation of unifying quotation marks and lowercase, we find that dealing with ambiguous data can improve MAR score 0.2%. On the other hand, data augmentation cannot improve the score. This may because text augmented by nlpaug may not match with their labels. Finally, we adjust the training and validation ratio into 0.95, which means that there are more data in the training set, and we obtain the MAR score of 0.5151.

Data	preprocessing	model	MAR
text	-	BERT	0.4949
text+reply	-	BERT	0.5008
text+reply	Q+L	BERT	0.5063
text+reply	Q+E+L	BERT	0.5107
text+reply	Q+E+A+L	BERT	0.5094
text+reply	Q+E+L	BERT*	0.5151
text+reply (ambi_data)	Q+L	BERT	0.5127
text+reply (ambi_data)	Q+L	BERT*	0.5036
text+reply (augmentation)	Q+L	BERT	0.4937

Table 2: Ablation study. ** means train:validation = 95:5. Q means preprocessing quotation mark. E means Demojizing, A means replacing abbreviations. L means turning tweet sentences in lowercase.*

Model	MAR score
BERT-Base	0.5127
BERT-Large	0.53925
RoBERTa-Base	0.5520
RoBERTa-Large	0.5783

Table 3: Performance of BERT and RoBERTa

4.4.2 BERT and RoBERTa

To further improve the performance, we take RoBERTa as the pre-trained model which we apply transfer learning from and compare its performance with the BERT one. We also try different scales of pre-trained models, such as BERT-Base, BERT-Large, RoBERTa-Base, and RoBERTa-Large. As shown in Table 3, both the large version of these two models outperform the base version. And the model based on pre-trained RoBERTa-Large achieves the highest performance in our work with a score of 0.5783 on development data.

5 Conclusion

In this work, we present a multi-label classification model that utilizes the strength of pre-trained models to predict GIF responses. We use preprocessing for tweets and then fine-tune two kinds of pre-trained models to handle the specific task in the EmotionGIF challenge. We find that concatenating reply to text can improve the performance a lot but not every preprocessing method is suitable for this task. Although we try to mitigate the effect of the class-imbalance problem, our augmentation method does not work well. At final, our RoBERTa-based model could achieve the MAR score of 0.5824 on the evaluation data.

References

Carla Bazzanella. Emotions, language and context. *Emotion in dialogic interaction: Advances in the Complex*, pages 55–72, 2004.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

Weiyuan Li and Hua Xu. Text-based emotion classification using emotion cause extraction. *Expert Systems with Applications*, 41(4):1742–1749, 2014.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification?, 2019.