Emotion Classification on Empathetic Dialogues using BERT-Based Models

Hung-Yi Wu **Tzu-Jui Chen** **Chun-Kang Kung**
Department of Computer Science
National Yang Ming Chiao Tung University

1 Introduction

Natural Language Processing (NLP) is the field concerned with the interactions between computers and human languages, and, concerned with helping computers to understand and process human (natural) languages. The researches about NLP has began from 1950s and it is still an extremely popular research field until today due to the complexity and uncertainty of human languages. In the lectures, we have learned many traditional methods to help us analyze and process corpus such as parsing tree, Probabilistic Context Free Grammars (PCFG), Chomsky Normal Form (CNF). Though these methods are often adopted in NLP related experiments nowadays, they are often used as supplements only. In fact, in these days, the main method to resolve an NLP work, is by using Transformer. The development of Transformer is a breakthrough in NLP field, it's high accuracy and wide generalization surprised every expert in this field. Based on the architecture of Transformer, more and more extraordinary model has been developed and the abilities for computers' to understand human languages in some tasks has become really close to humans' abilities. In this experiment, we use one of the most famous and widely-used transformer based models, BERT (Devlin et al., 2018) and RoBERTa (Zhuang et al., 2021) to predict the emotion from a conversation and its preset emotional condition. In addition, we tried to pre-process our data in traditional NLP methods to improve our final results. In the end, we did a complete experiment to evaluate and compare our results with others.

2 Related Work

Natural Language Processing (NLP) is the field for computers to interpret, analyze, and learn meanings from text, conversations and languages. At first, individual NLP tasks were traditionally solved by individual and specific models since the complicate differences between each NLP task. However, in 2018, a revolutionary model - BERT came to the world and completely change the development of NLP. The original code of Transformer and BERT (Devlin et al., 2018) are both developed by Google Research team. Fortunately, the NLP, AI communities have spent a lot of time on building open-source research tools for researchers. Through communities' hard working, more and more libraries are built to provide user-friendly models for easy downloading, caching, and fine-tuning. One of the core library we used in our project is Hugging Face's (Wolf et al., 2019). This platform is one of the most popular platform when it comes to AI and NLP. It provides researchers to build, train and deploy models smoothly and easily. In this project, we benefit from the variety of BERT models that Hugging face provides, such as the large, base BERT and RoBERTa (Zhuang et al., 2021) models. Different models have different numbers of layers, attention heads, blocks to apply on different tasks. Furthermore, the Hugging Face library also provide user-friendly tokenizer to help people complete their data processing progress.

3 Methods

3.1 Dataset

Empathetic Dialogues (ED): This dataset was constructed by (Rashkin et al., 2018) in 2018. It contains about 24k conversations, 50k utterances, and the average utterance per conversation is 4.31. The data is collected from 810 different crowd workers' conversations each grounded in an emotional situation. In each dialogue, the conversation always starts from a pre-defined emotional condition (e.g., happy, sad). One person acts as a speaker who describes his or her situation. Meanwhile, the others plays a "listener" role, displaying empathy during

the discussion and understanding the underlying emotional connections between what the speaker says and responds. After several testing, this corpus is proved to be useful in helping the model to understand and learn the empathy in conversations.

3.2 Data Preprocessing

The main goal of this part is to utilize Natural Language Processing techniques to cleanse the input data, remove the noises and prepare for the next step and hope the model can extract proper and useful features.

The dataset we use contains prompts and utterances. These two types of texts need to be preprocessed through a data preprocessing pipeline. This data preprocessing pipeline includes the following 4 parts:

1. Replace characters that are not between a to z or A to Z with white space.
2. Convert all characters into lower-case ones.
3. Remove the inflectional morphemes like "ed", "est", "s", and "ing" from their token stem. Ex: confirmed → "confirm" + "-ed".
4. Replace "_comma_" with " , " in the sentences.

3.3 Data Manipulation Techniques

In this project, we attempted to use multiple different data manipulation techniques to improve the classification performance. We proposed 3 different techniques and the details are as follows.

- **Grouping** In this dataset, there are multiple utterances that belongs to a prompt (conversation). After careful experiments, we believe that training with such data may cause conflicted classification, which the model may produce different classification on utterances belonging to the same prompt. Therefore, we decide to group the utterances which belongs to the same prompt into an entry in the dataset in order to leverage the prompt feature and avoid the conflicted classification problem. We implement this idea by concatenating the utterances with the same prompt together and form a new data with the prompt and the label.
- **FastText** FastText is an open-source, light weight library to help user learn text representation and text classifier. We expected that by adding additional information (emotional tags) from a pretrain classifier, the final accuracy can be improved. The pretrained model we used is referenced from (Rashkin et al., 2019) and was pretrained by Facebook's research teams. This model can predict single or multiple emotional labels fast and accurately by sending a sentence as an input. Furthermore, the model's predict classes (ex: nostalgic, devastated) are different from the classes we want to predict in our project (ex: afraid, happy). Therefore, we hope it can provide some additional information which is learned in other datasets as data augmentation. In our project, we tried to concatenate emotional labels which are predicted by Fasttext classifier in the beginning of prompts and utterances. In this way, our model can utilize the extra information to help the overall predictions.
- **Role Feature** The *Empathetic Dialogues* dataset contains prompts and utterances, and each unique prompt are dependent to multiple utterances. We believe that the speaker of an utterance will affect the meaning of the conversation to a certain level, so we decide to add an extra feature to represent the characteristic of the speaker and we define this feature as 'Role Feature'. In this project, we implement this extra feature by prepending 'A says' and 'B says' to the utterances in turn in each conversation to form a scenario that two people are having a conversation and they have different characteristics.

3.4 Emotion Classification on Empathetic Dialogues

Over the past few years, models that based on the idea of BERT have flourished in the field of Natural Language Processing. In this project, we choose BERT-based models, BERT and RoBERTa, to perform the *Emotion Classification on Empathetic Dialogues*. However, they are slightly different in embeddings and pretraining steps.

3.5 Word Embedding

For BERT, we use BERT tokenizer to tokenize the prompts and the utterances in the dataset. BERT uses Google NMT's WordPiece Tokenization [11] to separate the words into smaller pieces to deal with words that are not contained in the dictionary. For example, embedding is divided into ['em', '##bed', '##ding'].

Also, there are two special tokens that BERT needs, which are [CLS] and [SEP]. [CLS] is put at the beginning of a sentence representing the front of an input series. [SEP] is put at the middle of two groups of sentences when they are combined to a single input series, and is put at the end of the input series as well to mark the end of an input sequence.

Then, for the input of BERT, we need to convert the original statements into three kinds of tensors, which are token tensors, segment tensors, and mask tensors. Token tensors represent the indices of tokens, which are obtained by the tokenizer. Segment tensors represent the identification of different groups of sentences. Mask tensors represent the concentration of tokens including information after zero-padding the data into sequences of the same length.

For RoBERTa, we use Byte-Pair Encoding (BPE) (Sennrich et al., 2016) as the representation of text encoding. BPE is a hybrid between character- and word-level representations that allows handling the large vocabularies common in natural language corpora. Instead of full words, BPE relies on subwords units, which are extracted by performing statistical analysis of the training corpus.

BPE vocabulary sizes typically range from 10K-100K subword units. However, unicode characters can account for a sizeable portion of this vocabulary when modeling large and diverse corpora, such as the ones considered in this work. (Radford et al., 2019) introduce a clever implementation of BPE that uses bytes instead of unicode characters as the base subword units. Using bytes makes it possible to learn a subword vocabulary of a modest size (50K units) that can still encode any input text without introducing any "unknown" tokens.

3.6 BERT-Based Models

BERT, Bidirectional Encoder Representations from Transformers, is a transformer-based machine learning technique that changed the NLP world in recent years due to its state-of-the-art performance. Its two main features are that it is a deep transformer model so that it can process lengthy sentences effectively using the 'attention' mechanism, and it is bidirectional so that it will output based on the entire input sentence. We used BERT to handle the dataset and construct a deep learning model by fine-tuning the *bert-based-uncased* pretrained model for *Emotion Classification on Empathetic Dialogues*.

Training a model for natural language processing is costly and time-consuming because of the large number of parameters. Fortunately, we have pretrained models of BERT that enable us to conduct transfer learning efficiently. We choose the pretrained model of *bert-base-uncased* from a lot of models with different kinds of parameters. The chosen one consists of a base amount of parameters and does not consider cases of letters (upper-case and lower-case).

For different downstream tasks, we need to conduct different fine-tuning approaches. Thanks to HuggingFace, we have the models for different downstream tasks. In our project of emotion classification, we used *bertForSequenceClassification* to fine-tune our pre-trained BERT model. The modules of the model contain a BERT module handling various embeddings, a BERT transformer encoder, a *BertPooler*, a *Dropout* layer, and a linear classifier that returns logits of the 32 classes, which is the total number of emotions in the dataset.

RoBERTa, a Robustly Optimized BERT Pretraining Approach, is the most successful variant of the BERT pretraining procedure that improve end-task performance. Specifically, RoBERTa is trained with dynamic masking, FULL-SENTENCES without NSP loss, large mini-batches and a larger byte-level BPE. The specific explaination of 'dynamic masking' and 'FULL-SENTENCES without NSP loss' are as follows.

BERT relies on randomly masking and predicting tokens. The original BERT implementation performed masking once during data preprocessing, resulting in a single static mask. On the other hand, RoBERTa relies on dynamic masking where it generate the masking pattern every time the model is fed with a sequence. The author of RoBERTa found that their reimplementation with static masking performs similar to the original BERT model, and dynamic masking is comparable or slightly better than static masking.

In the original BERT pretraining procedure, the model is trained to predict whether the observed document segments come from the same or distinct documents via an auxiliary Next Sentence Prediction (NSP) loss. The NSP loss was hypothesized to be an important factor in training the original BERT model. However, some recent work has questioned the necessity of the NSP loss (Lample and Conneau, 2019; Yang et al., 2019; Joshi et al.,

2020). (Zhuang et al., 2021) compared several alternative training formats (SEGMENT-PAIR+NSP, SENTENCE-PAIR+NSP, FULL-SENTENCES, DOC-SENTENCES) to better understand this discrepancy. They found that removing the NSP loss matches improves downstream task performance. Finally they found that restricting sequences to come from a single document (DOC-SENTENCES) performs slightly better than packing sequences from multiple documents (FULL-SENTENCES). However, because the DOC-SENTENCES format results in variable batch sizes, we use FULL-SENTENCES in the remainder of our experiments for easier comparison with related work.

4 Experiment

In this project, we used multiple different methods to find the best performance of *Emotion Classification on Empathetic Dialogues*. Table 1 shows the performance of our methods. What stands out in the table is that using concatenate utterance and role feature got little improvement for the performance, but using the grouped data and RoBERTa made marked progress.

It is apparent from this table that using the grouped data method with concatenating utterance would get better performance. The performance score of our experiment increase from 0.564 to 0.599 after we change the data to concatenated utterances.The score is from NLP 110 Final Project - Basic (Mandatory) on Kaggle. Furthermore, The performance score of our experiment increase from 0.599 to 0.625 after we change the model from BERT to RoBERTa. However, there is no significant difference between using role feature or not. The performance score of our experiment only increase for 0.005. We think that it is because the grouped data can't really separate into two role feature because most of them have nothing do with each other.

5 Conclusion

In this project, we tried many different methods to perform the *Emotion Classification on Empathetic Dialogues*. First, we use simple data with BERT. To improve the performance of the classifier, we propose 3 data manipulation techniques (Grouping, FastText and Role Feature) on our basic structure. Among these, the Grouping method substantially increased the prediction score, but the FastText and Role Feature methods improved the performance slightly. To further enhance our model, we convert our model from BERT to RoBERTa and finally got a leap on the performance. Therefore, we chose Grouped Data with RoBERTa as our final proposed classification method of *Emotion Classification on Empathetic Dialogues*.

Model	Data	RF	Score
BERT	prompt	-	0.557
BERT	prompt & utterance	-	0.564
BERT	grouped data	-	0.599
BERT	prompt & utterance	v	0.569
BERT	Fasttext&grouped data	-	0.573
BERT	grouped data	v	0.592
RoBERTa	grouped data	-	**0.625**

Table 1: The performance of our methods of Emotion Classification on Empathetic Dialogues. We tried different data handling methods such as grouping data by prompts, adding role features as extra labels, augmenting data with Fasttext. Furthermore, we tried different models in this task. Eventually, we found that using RoBERTa and grouped data outperforms other combinations. **RF** stands for Role Feature.

6 Work Division

- **Hung-Yi Wu:** Built prompt, prompt + utterance, grouped data, and role features with BERT
- **Tzu-Jui Chen:** Surveyed on FastText and applied it to prompt, prompt + utterance, and grouped data with BERT
- **Chun-Kang Kung:** Surveyed on RoBERTa and applied our methods to it

7 Question and Answer

1. **Question**
   想請問這個加在前面的emotion tag跟實際label的吻合程度？

   Answer
   加在前面的emotion tag與實際label是獨立的，emotion tag是利用另一個模型預測出來的結果，並非基於dataset提供的label。

2. **Question**
   請問有嘗試做Freeze克服GPU memory不足的問題嗎？

   Answer
   沒有。

3. **Question**
   要嘗試過把label mapping成sent嗎？

   Answer
   沒有。

4. **Question**
   想請問在training加入label tag，但testing並沒有label資訊。想請問加入的想法是甚麼呢？

   Answer
   在testing時，一樣會將testing data加上emotion tag再進行預測。

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2018. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: a new benchmark and dataset. In *ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Association for Computational Linguistics (ACL)*, pages 1715–1725.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized bert pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227.