|  |
| --- |
| **BERT-based and XLNET-based Empathetic Dialogues** |
| **Multiclass Sentiment Analysis** |
| **Chih-Chao, Hsu.** |
|  |
|  |
|  |
|  |

Introduction

Nowadays, with most kinds of markets become more and more competitive, it is crucial that a company understand their customers' sentiments, which including what they are saying, what they mean and how they are conveying. Attempting to seize the true emotions behind the customers' reviews, tweets, or comments on social media platforms can help a company knows not only the opinions regarding their brands, but also what are the things that their users genuinely concern (Westerski, 2007). Apart from the ordinary reviews, the amounts of commentary in the form of dialogues have increased rapidly in recent years, thanked to how wide-spread and convenient the social media platforms are. Thus, sentiment classification in dialogues quickly becomes a popular and challenging task. Moreover, humans tend to have multiple emotions while expressing their thoughts and feelings. Therefore, choosing the correct sentiment behind utterances of dialogue among several candidate emotions is particularly complex and interesting (Firdaus et al., 2020).

In this paper, we would like to utilize the well-known BERT model (Devlin et al., 2018) and an improvised model called XLNet model (Yang et al., 2019) to conduct multiclass sentiment analysis on a public empathetic dialogue dataset (Rashkin et al., 2018). Besides from examining the performances of these 2 models, we would like to test the effectiveness of some data pre-processing trick, such as removing the punctuations and stop words, lemmatization, and data augmentation via back translation. Our results will be presented in the fashion of macro-F1 score trend on the testing dataset with several worth-mentioning milestone.

Related Work

In this section, we have reviewed some papers regarding some data pre-processing tricks and models to be used. Based on the outline of our research, we would like to split these parts into 3 little sections: **data pre-processing, BERT model,** and **XLNet model.**

**Data pre-processing**: It is a common practice to remove stop words and punctuations in lots of text classification related tasks. Many previous researchers generally removed the punctuation marks in their studies (Akba et al., 2014; Cetin et al., 2013; Demirtas et al., 2013; Kaya et al., 2012), as these punctuations often interfere the word tokenizing processes. However, as the results Kaya et al. (2012) experimented, sometimes the impacts of keeping punctuation marks as a textual feature in some languages such as Turkish are worth discussing.

As for removing the stop words, Ladani and Desai (2020) mentioned that the benefits of removing stop words include decrease in size of corpus, improvement of efficiency and accuracy of the text mining applications thus helping in reduction of time and space complexity of overall application. Other reasons suggested by Ladani and Desai are that stop words have low discrimination power, and they have no meaning, no predictive ability but high frequency of occurrence in the text, which generally make them noises when doing the text classification tasks.

We also considered applying lemmatization on the datasets, as there are researchers (Toman et al., 2006) stated that lemmatization in some cases can be beneficial, since it produces the basic word form which is required in many application areas.

Last but not least, Ma and Li (2020) have argued that when the amount of data is insufficient, or the distribution of samples is unbalanced, the accuracy of text classification will be greatly affected. Data augmentation is a necessary action in these scenarios. As for why back-translation might be a good data augmentation technique, it is because that back-translation can generate diverse paraphrases while preserving the semantics of the original sentences (Ma and Li, 2020).

Method

Same as the order described in the previous "Related Works" section, based on the outline of our research, we would like to split these parts into 3 little sections: **data pre-processing, BERT model,** and **XLNet model.**

**Data pre-processing**: The empathetic dialogue datasets include a prompt and several utterances for each conversation. Besides those, there is a numeric label indicating the ground truth sentiment label for each dialogue. There are totally 32 candidate emotions in the datasets. Both training dataset and validation dataset, but not testing dataset, have these labels.

The first step of data pre-processing is to remove punctuation marks. We removed all period dots ".", "\_comma\_" symbols which are assumed to be comma originally, exclamation marks "!" and question marks "?". Upon checking the first few samples in the training dataset, we also found out that there exist some emojis, such as ":(". So, we cleaned those, too. After that, we got rid of the leading spaces and trailing spaces, as well as the new line character "\n" at the end of each sentences. Finally, since we would like to try the uncased pretrained BERT model later on, we lowered cases of all English letters.

The next step is to delete stop words and to execute lemmatization on all the remaining words. These 2 pre-processing are done by the aid of NLTK package (Bird et al., 2009), which is a prominent natural language toolkit. According to the official documentation, there are 40 stop words in the stop words list. We filtered them out. After that, we did the lemmatization for each word remaining, based on their part-of-speech tagging, respectively. Lemmatization removes the suffix of a word completely to get the basic word form or replaces the suffix of a word with a different one. Since in NLTK package, the WordNetLemmatizer uses some built-in function from the WordNet (Fellbaum, 1998), a large and publicly available lexical database in English, this lemmatization step would only be executed in "advanced" task. Furthermore, note that we basically conducted these NLTK pre-processing on utterances, not on prompt. We would, however, also try these 2 pre-processing techniques on prompts later in our implementation as well, for showing how prompts play an important role in empathetic classification.

All these pre-processing tricks are wrapped in a function. We called the function to clean the datasets before loading them in data loaders.

We have also tried data augmentation in the "advanced" task, by back-translation. About the implementation details, we call the Google translate API (Han, 2015) to translate prompts. In our practice, we randomly choose 5 samples from each of the 32 emotion labels. And then, randomly translate them to 1 of the 10 target languages we chose, and then translate them back. The 10 target languages are Chinese (both simplified and traditional), Japanese, Korean, French, Germen, Spanish, Indonesian, Arabic and Russian. The reason we chose these 10 languages is that they are high-resource languages with many users.

Note that we only tried to back-translate prompts, not utterances in this case. This is owing to that utterances have lost grammatical structure after deleting stop words and doing lemmatization. It is inappropriate to back-translate these obviously broken sentences.

Experiment

Conclusion

File Format

References (留一個當參考格式not ours)

Alfred. V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling, volume 1*. Prentice-Hall, Englewood Cliffs, NJ.

1. Work Division
2. Question and Answer

**Question 1** (Professor Gu): In terms of model architecture, why do you think the XLNet would have better performance than BERT did?

**Answer 1**: Although similar to BERT model, XLNet does outperform BERT in roughly 20 NLP tasks, and sometimes with quite substantial margins (Kanishk, 2020). Also, as Adoma et al. (2020) pointed out in their study, XLNet does have better performance than BERT does in text-based emotion recognition, which is similar to our task. The major differences between these 2 models come in the approaches to pre-training.

Basically, BERT is an autoencoding based model, while XLNet is an autoregressive based model. This difference materializes in the masked language model (MLM) tasks, where randomly masked language tokens are to be predicted by the model. By the help of a technique called permutation language modeling (PLM), XLNet can learn more contextual information without assuming the independence between masked tokens as the BERT model does. Thus, XLNet is able to capture the dependency between the masked tokens.

**Question 2** (student 0810749): Why do you think deleting punctuation marks would result in better performance? Is there any intuitional explanation?

**Answer 2**: Because we would tokenize the sentences before feeding them to the models, we would like to remove these punctuation marks. We found out that no matter the BERT tokenizer or the XLNet tokenizer, these tokenizers deem these punctuations as tokens as well. These punctuation tokens are just meaningless noises when performing contextual recognitions. The experimental results indeed indicated better performance when removing punctuations.

**Question 3** (student 0716050): Why do you set the layer number of your downstream classifier as 4 layers initially? And why do you reduce to 1 layer later on?

**Answer 3**: Actually, the layer number of a classifier should be considered relative to the difficulty of the classification tasks. As a multiclass classification problem with up to 32 kinds of labels, we initially set up 4 layers as suggested in a paper with similar classification problem (Sahoo et al., 2020). However, later on we found out that as simple as it should be, only 1 layer of downstream classifier is enough for generating good classification results, referred to the study conducted by original BERT team (Devlin et al., 2018) regarding sentence pair classification. We’ve tried using only 1 layer, and found out that the performance holds roughly the same. Thus, for simplicity principle, we kept this setting in our later experiments.