IR&TM Project 2 Documentation

Bouruc Petru-Liviu
petru.bouruc@s.unibuc.ro

Abstract

This is the documentation for the second project at Information Retrieval & Text Mining. It contains the tried approaches for both tasks and their results.

1 Introduction

For this project we had to solve 2 subtasks on a given dataset composed of conversations between different people, where each conversation had utterances. Each utterance expressed an emotion (7 classes). I used $train\_dataset$ and $test\_dataset$ in order to solve the tasks.

2 Engines

In order to solve the given tasks, I used $pandas$ library to read the data, $nltk$ library to preprocess the data and $sklearn$ modules with classical models and evaluation metrics. For the deep approaches, I used $tensorflow$ given models.

In order to train the BERT model I used Google Colab for the extra GPU power it provides.

3 Document Body

3.1 Task 0

For the first task where we had to classify each utterance into a class representing an emotion, I had 3 approaches: classical machine learning models, deeplearning model (a RNN model - BiLSTM) and using a transformer (BERT).

I tried different **text preprocessing** approaches. I observed that with much preprocessing, the accuracy drops. If I do word tokenize, remove stop words and then apply stemming, the highest accuracy I could obtain was 0.51. This effect may be cause by the fact that some stop words or signs can put a greater emphasis on some emotions. By keeping only the stemming, I managed to get better results.

kernel	**Output**
rbf	1
linear	3.5
	10

Table 1: Grid for SVM classifier

n_estimators	**max_depth**
50	2
100	4
150	6
200	8
	None

Table 2: Grid for RandomForest classifier

For the **feature extractor and models**, firstly I tried a combination of TF-IDF and Bag of Words (BoW), and for the SVM and RandomForest I applied a **GridSearch** with the hyperparams that can be seen in Table 1 and Table 2.

I noted down the best results from the both grids in Table 3 containing all results.

As for the deeplearning model, I tried to use a **Recurent Neural Network (BiLSTM)** with Word2Vec features. I used $ReduceOnPlateau$ callback and an early stopping in order to prevent overfitting . The best results I got hypertuning the $embedding\_dim$ hyperparameter was 0.55.

The last approach I tried was to fine-tune a **BERT transformer**. After the first epoch I obtained an accuracy of about 0.61 on the test. I saved the model and continue to train. On the second epoch the accuracy dropped a bit, so the best result was to use the weights from the first epoch train.

Overall results can be seen in Table 3.

3.2 Task 1

For the second task, I tried to do some research on how it can be done. I stumbled upon Multi-

Feature Extractor	**Model**	**Hyperparameters**	**Acc | F1**
TF-IDF	Naive-Bayes		0.57 | 0.54
BoW	Naive-Bayes		0.49 | 0.37
TF-IDF	SVM	C=3.5 kernel='rbf'	0.57 | 0.54
BoW	SVM	C=3.5 kernel='rbf'	0.59 | 0.55
TF-IDF	RandomForest	n_estimators=100 max_depth='None'	0.57 | 0.52
BoW	RandomForest	n_estimators=200 max_depth='None'	0.57 | 0.52
Word2Vec	BiLSTM	embedding_dim = 30	0.56
Word2Vec	BiLSTM	embedding_dim = 100	0.52
BertTokenizer	BERT		0.61

Table 3: Feature extraction techniques, models, hyperparameter tuning and their respective accuracies.

modal Emotion-Cause Pair Extraction in Conversations paper where it shows how this task can be approached.

I also tried to do some prompt engineering with ChatGPT, but I didn't manage to get the prompts right in order to give me concludent results. It only classified the utterances.

I could not replicate or do such thing, so I tried a simpler approach to see how could a model would perform to predict a general emotion-cause pair (not including spans). I fitted the given emotion-cause pairs to a ML model and emotion-cause pairs made by pairing two non-related utterances from conversations and tried to predict on the test set if two utterances are related or not. I tried a combination of TF-IDF and BoW feature extractors and Naive-Bayes, RandomForest and SVM models. Regarding of the feature extractors used, both the RandomForest and SVM model made predictions with an accuracy of 0.71 or 0.72 and 0.70 or 0.71 weighted F1, while the Naive-Bayes model obtained an accuracy of 0.67 and 0.66 weighted F1.

4 Conclusions

The Emotion-Cause pairing is a recent problem which for sure can be improved in the future, as shown in the before metioned paper. As for the emotion classifier, the transformers seem to do the best job, but it would be interesting to see how LLMs can perform.