DSBA/CENTRALESUPELEC NLP_COURSE - NLP ASSIGNEMENT - 21/04/2023

----------NAMES----------

DARGIER Antoine - DOUILLY Thomas - RAMAMONJY Johary

----------METHODOLOGY----------

0) We started by reading scientific research papers on the subject [1], [2], to understand the issues, the classical methods and to get relevant ideas for the assignment. With these papers and our courses, we decided to fine-tune a pre-trained BERT model on unlabelled data. This is also what HuggingFace offers on their site, with example notebooks to show how to fine-tune their models [3]. We were inspired by this example.

1) Data pre-processing: To be able to use our data, we first had to transform it into usable data. We converted the polarities into numerical values (0: negative, 1: neutral, 2: positive). To take into account the study aspects and words, we transformed and joined them. The method we have chosen is to formulate these sentences as follows: "Feeling about WORD ASPECT". For example, for a category "FOOD#PRICES" and a word "meat", this will give: "Feeling about meat prices". We tried to phrase these sentences in a number of different ways, in the form of a question, and this seemed to be the most effective method.

2) Tokenization: As advised, we used a pre-trained tokenizer to create the embeddings of these aspects and phrases with context. So we used BertTokenizer to create the embeddings. Using the tokenize_plus() function we were able to generate the embeddings for the aspect and the sentence at the same time. In this function we used padding to fill in shorter sentences with 0's, and set the length to 20% longer than the maximum length in the train set. This function returns the input_ids and attention_mask, which indicates which elements of the input_ids sequence should be considered for the classification task. This also allows the elements created by the padding to be targeted.

3) Model: For the model, we used a 'bert-base-uncased' BERT model which we adapted by TransferLearning. We added two layers to the model: a dropout layer to limit overfitting, and a linear layer to have three categories as output. We then train the model on the train set, and look at the performance on the dev set to keep the best model on the dev set. We choose to use the Adam optimizer with a linear learning rate scheduler, which seemed to be the most used for this kind of model.

4) Fine-Tuning: Three parameters were important to fine-tune and play a very important role in the quality of the classification: the dropout rate, the learning rate of the optimizer and the number of epochs. We started by setting the dropout rate to 0.30 after multiple trials. This gave the best results. Then we had to find a balance between learning rate and epochs, to be sure to reach convergence and the best results, without losing the speed of our method. We therefore chose to take a learning rate of 8e-6, with 8 epochs, which ensure convergence. Indeed, the best prediction values are obtained around 3-4 epochs, before the model overfits the train set much more.

5) Prediction: We logically used the softmax and argmax functions to retrieve the predominant class. And we converted the outputs to text again, in the opposite direction to the pre-processing.

----------TIME OF TRAINING----------

Time for training and inference: Exec time: 1288.02 s. (257 per run), on a Google Colab environment

----------RESULTS----------

Our accuracy on the dev set: Completed 5 runs.

Dev accs: [87.23, 85.37, 87.23, 86.44, 85.11]. Mean Dev Acc.: 86.28 (0.90)

----------RESSOURCES----------

[1] W. Zhang, X. Li, Y. Deng, L. Bing, W. Lam, "A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges", 2022.

[2] H. Xu, B. Liu, L. Shu, and P. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis" in NAACL-HLT, 2019, pp. 2324–2335.

[3] https://colab.research.google.com/github/DhavalTaunk08/Transformers_scripts/blob/master/Transformers_multilabel_distilbert.ipynb#scrollTo=yU4TWUBtNKUN