






Attachments:

-  BERTmultilingual.ipynb
-  DistilBERT.ipynb
-  parsBERT.ipynb
-  prepare_data.ipynb
-  TextAttack.ipynb

The first step: I divide the persica.csv data into test and training sets in a ratio of 8 to 2. The data that is examined as the text is the combination of the title and the text of the news, which was also used in the previous exercise. Now, among the training data, I will separate the ratio of 2 to 8 for the validation data in the process of training and adjusting the weights. Because the training of each network requires a lot of time and to avoid repeating this division every time, I saved the divided data in three files: train.csv, test.csv, and validation.csv, and this The files are used for fine-tuning all the models used in this report. Columns from the main persica.csv file that were unrelated to the exercise did not appear in these files either. This is because the output of the networks is the probabilities that apply to each label, for the convenience of work, I converted the labels into numbers so that each probability index number is equal to it in the output vector of the network, both in terms of concept and in terms of Validation will be easier. This operation is done in prepare_data.ipynb file.

The second step: selecting the networks that are going to be used and fine-tuning them into 11 separate classes to categorize the news. I used 4 BERT models. All models are linguistic models, which are networks that have been trained by masking a part of their inputs and then predicting the masked part in a self-supervised manner and have not yet been fine-tuned for any application. The models are selected from <https://huggingface.co/>.

- textattack/bert-base-uncased-ag-news: A network trained with English news data and its fine-tuned models have provided good results for text classification according to reports.
- distilbert-base-uncased: a model trained with data from Wikipedia and English books and magazines.
- HooshvareLab/bert-fa-zwnj-base: It is the same model as Persian bert or parsbert.
- bert-base-multilingual-cased : A language model trained with data from 104 different languages.

The next steps are done for each of these models in a separate file. These files were implemented on colab and due to time constraints for using resources, they are not all in one file.

Third step: I open the prepared files as dataset classes so that pre-processing can be done on it. The input text will be standardized using the selected grid. In this operation, the number of inputs is adjusted to the size of the network opening, margins are added or subtracted from it, and each attack is divided into its constituent words, and the

output of this stage includes data that has been fully pre-processed. and are ready to enter the primary layer of the network. The encoding of positions and attention is calculated for the input layer.

The fourth step: adjusting the parameters for fine-tuning. For each of the networks, groups of 8 or 16 with IPACs of sizes 3 to 10 were selected. Of course, they usually consider the number 3 or 4 for this. I moved the learning rate between 1e-3 and 1e-5 in each case. And I fine-tuned the networks many times to get better results. I brought the best model obtained from each in the files. Due to the large volume of models, only the code containing the result of their implementation is attached. The results that are reported below are calculated based on the learning rate of 0.00001 and with 3 IPACs. In the case of parsbert, this was satisfactory, but in the case of the rest of the models, there was no significant change in the model's performance with the changes I made in the parameters. The other three models did not show correct performance on the data at all, and the loss value did not show a clear sign of convergence even in the 10th to 15th rounds.

Fifth step: After the fine-tuning is finished, I calculated the precision, recall, f1 value for the test data as in the previous exercise. which is given below.

	PRECISION	RECALL	F1 SCORE
textattack/bert-base-uncased-ag-news	9	1	1
distilbert-base-uncased	14	4	5
HooshvareLab/bert-fa-zwnj-base	89	89	89
bert-base-multilingual-cased	9	1	1

As mentioned before, the only acceptable and usable model here is parsbert. The rest of the models probably had problems modeling the Persian language structure. Or, for example, they have not been able to tokenize the input words in the desired way. In all executions, the output of the model is that it outputs a label for all test data, in fact there is no classification at all.

Now I bring the results of the previous exercise below:

	Naive_Bayes	Perceptron	SVM
precision	0.77	0.76	0.82
recall	0.78	0.78	0.82
f-score	0.77	0.76	0.82

All the three classical models obtained acceptable results on the tf-idf matrix of the previous exercise on which lsa transformation was performed, but the SVM model obtained better results. Compare this model with parsbert. The use of a suitable Bert network has significantly improved the validation criteria.