Assignment 2

Morotti Daniele, Sciamarelli Andrea and **Valente Andrea**
Master's Degree in Artificial Intelligence, University of Bologna
{ daniele.morotti, andrea.sciamarelli, andrea.valente6 }@studio.unibo.it

Abstract

The objective of this assignment is performing Question Answering on (Siva et al., 2018) CoQA dataset using transformer-based architectures with Bert-like models for both encoder and decoder parts. There are different types of QA tasks. In our case, given a question and the relative text passage, the model should be able to generate an answer. We defined the different models using the Huggingface library (`https://huggingface.co/`) on top of Tensorflow and Keras. Moreover, we had to test multiple models, also considering the history together with the text passage. The history is composed, for each sample, of the text passage and the previous question-answer pairs within the same dialogue.

1 Introduction

Considering QA tasks, most of the models that can be found in the literature tackle the problem with an extractive approach. This means that the predicted answers are directly found in the text passage and the model is trained using 2 values that represent the beginning of the answer and the end of it. A more difficult problem is when you have to generate an answer given a question and the paired context, as in our assignment. Differently from other QA datasets, questions in CoQA are conversational, namely each question after the first depends on the previous ones. One of the SOTA models (Ju et al., 2019) on CoQA is trained with adversarial training and knowledge distillation using RoBERTa as baseline. Some other models used T5-like architectures that are more powerful and more suitable for text generation because of its nature (it is an encoder-decoder transformer).

2 System description

In this assignment we had to use TinyBERT and DistilRoBERTa as encoder and decoder parts of our architecture, a sequence to sequence model. In order to implement the models, we used the Huggingface module *TFEncoderDecoderModel* that imports the desired pretrained checkpoints. Before training the models, we needed the relative tokenizers that compute the vector of tokens and the attention mask, given an input string. TinyBERT and DistilRoBERTa use different tokenization processes, therefore we imported the two tokenizers considering the model we were training. The maximum number of input tokens is 512 in both BERT's alternative and we decided to use this value also for our data, trying to exploit as much information as possible, given that it was mandatory to train for 3 epochs. We had to train 12 models considering several combinations of settings, that include the choice between TinyBERT and RoBERTa; whether to use the conversational history or not; the seeds used while training (i.e. 42, 1337 and 2022), for reproducibility and also to ensure that the performances are not influenced by the choice of a particular random seed. Moreover we had to remove the unanswerable questions in order to simplify the whole assignment.

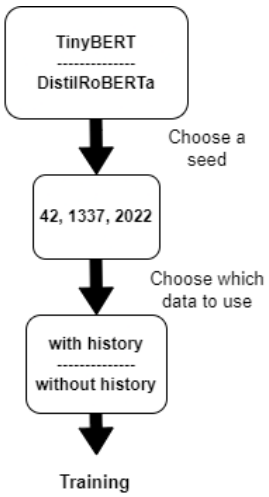

Figure 1: Different model combinations

3 Experimental setup and results

All the code has been implemented using Huggingface, Keras, Tensorflow and the experiments were run on Google Colab. Using the predefined schema shown before, we created 12 different models and we collected the results in some external files in order to speed up the successive tests and comparisons. We didn't have to set a lot of hyperparameters, the only ones that we chose are the max size for the input embedding, the max size for the ground truth embedding, the batch size and the learning rate. As specified in the task statement, the training set was split in training and validation sets with a ratio of 80/20.

As it was mentioned before, we left the max size for the input data to 512 (i.e. the maximum one) but we decreased the one for the labels to 30 in order to reduce the computational load, also due to the fact that the length of almost all the answers does not exceed this value. We chose Adam algorithm as optimizer as it is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters" (Kingma and Ba, 2014). The learning rate was set to $3e-5$ because it's a typical value for the fine-tuning on BERT-like architectures. We left the default loss function without changing the one of the imported pre-trained models (i.e. cross-entropy loss). On the TinyBERT models, we performed the training with all the samples in the training set and we used all the validation samples, while on the DistilRoBERTa models we only used 10 thousand elements for the training and 2 thousand for the validation since, due to the high number of parameters, the process was computationally intensive.

For this task we had to use the SQuAD F1 score: this metric measures the average overlap between the prediction and ground truth answer, it treats the predictions and ground truth as bags of tokens, and computes their F1. We tested the networks with both the validation and test sets, and we measured the quality of the predictions by calculating the SQuAD f1-score on the entire test set, while we limited the computation on the validation set to the same number of sample of the test one in order to compare them fairly. In the table 1, a summary of our findings can be found.

4 Discussion

In the table 1 you can see that all the models achieved similar results. The best one is DistilRoBERTa, trained with seed 1337 and without history, with an F1 score of 12.59. These results are quite bad but it may be normal considering the few epochs of training and the smaller number of samples in the case of DistilRoBERTa. Even though all the models performed similarly, you can notice that TinyBERTs seem generally better and this is probably due to the training process, because these models have seen a larger number of examples. We also had noticed that the models performed worse on the validation set than on the test one and this is because, during the initial split of the dataset, some of the most difficult questions were assigned to the validation set. It could have been better if the split was made with a different seed for each model, but we decided to follow the given list of tasks for this assignment. Almost all models seem to perform worse when they are trained with the history. This behaviour can be also observed in the original paper of CoQA when they used a seq2seq model (check section 6.3 of (Siva et al., 2018)) and also, with other architectures, the gains are little in terms of F1 score if you consider more than one previous questions.

	f1_test	f1_val
distil-roberta_1337_no_hist	12.59	10.67
distil-roberta_2022_no_hist	12.59	11.25
bert-tiny_2022_no_hist	12.52	11.69
bert-tiny_42_no_hist	12.47	11.76
bert-tiny_42_hist	12.38	11.56
bert-tiny_2022_hist	12.37	11.48
bert-tiny_1337_hist	12.29	11.72
bert-tiny_1337_no_hist	12.27	11.48
distil-roberta_1337_hist	11.60	10.06
distil-roberta_2022_hist	11.59	10.43
distil-roberta_42_no_hist	11.33	10.42
distil-roberta_42_hist	10.95	9.84

Table 1: F1 scores on test and validation sets

5 Conclusion

The tested models didn't perform very well and the predicted answers sometimes are generated using the most common ones (e.g. yes or no) without considering the text passage. Sometimes the conversational history is important though, specifically in generic questions whereby the model has probably learnt the most common answer in the dataset.

In those cases we notice an improvement in the quality of the predicted answers. Some future improvements would be related to number of samples given to the DistilRoBERTa models, increase the number of epochs and use 2 different models for the architecture, a BERT-like model for the encoding part and a generation one for the decoder part.

References

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on conversational question answering.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Reddy Siva, Chen Danqi, and Manning D. Christopher. 2018. Coqa: A conversational question answering challenge.