Assignment 2

Francesca Boccardi, Luigi Podda, Matteo Nestola
Master's Degree in Artificial Intelligence, University of Bologna
{francesca.boccardi, luigi.podda, matteo.nestola}@studio.unibo.it

Abstract

The goal of this study is to implement two transformers-based models, employing BERTTiny and DistilRoBERTa, for performing a Generative Question Answering task on CoQA dataset. Since questions may be dependent on previous dialogue turns, two versions of each model have been implemented, which differ in whether or not taking into account the conversation history. According to SQuAD F1-score metric on the validation and test sets, experiments have yielded acceptable results only on DistilRoBERTa models, which generate much more accurate answers, still passage-related even when incorrect. BERTTiny instead often tends to predict wrong and non-contextualized responses.

1 Introduction

The aim of this study is to address a Generative Question Answering task employing two different sequence-to-sequence transformer-based models, built using BERTTiny (5) and DistilRoBERTa-base (6) both as encoder and decoder, which are lighter version of BERT (7) and RoBERTa (8) respectively. Experiments are runned on CoQA dataset (1), which contains 127k questions with answers, related to 8k text passages from different domains. For each *Q&A* pair is available a span of text supporting the answer, called rationale. Since almost half of CoQA questions refers back to conversational history, for each model two different versions have been implemented: the baseline, which given a question *Q* and a passage *P*, must provide an answer *A*, and its history version, which, for each question, also takes into account the conversation history *H*, concatenating it after *P*. After been fine-tuned with 3 different seeds, each model has been evaluated using the SQuAD F1-score (2) on the validation and the test sets. Only DistilRoBERTa models obtain acceptable results, showing to predict, even when incorrectly, at least plausible and passage-related answers.

2 System description

Starting from a transformer-based Encoder-Decoder architecture, two different types of models are built: BERTTiny-based, where BERTTiny is used both as encoder and decoder; DistilRoBERTa-based, where instead DistilRoBERTa-base is used both as encoder and decoder.
For each type of model, two different versions are implemented:

- The baseline $f_\theta(Q, P)$, which receives as input a question *Q* concatenated with the text passage *P* and generates an answer *A*
- The history version $f_\theta(Q, P, H)$, which receives as input a question *Q* concatenated with the text passage *P* and the conversation history *H*, i.e. the set of questions and answers of previous turns, in descending temporal order, generating an answer *A*.

To have a benchmark against which to properly reason on models performances, for both types of models a third version (upper bound version) is implemented, which receives as input a question *Q* and the rationale *R* rather than the passage.
Then, data is properly prepared. Since CoQA provides only two sets of data, containing 7193 and 500 passages respectively, the first one is divided into 80% for train and 20% for validation, while the second one is used as test set.
Before feeding the models, data are tokenized and transformed in model inputs, using a proper tokenizer. BERT and RoBERTa encoders accept input sequences maximum 512 tokens long, thus longer passages undergo truncation. Answers used as labels during the training phase are also tokenized or padded according to the maximum length computed as to handle the 90% of the training set answers, which is 9 tokens. Then, each model is fine-tuned (10), as to let them learn how to deal with the task to solve.

	seed: **42**		seed: **2022**		seed: **1337**	
	Val	**Test**	**Val**	**Test**	**Val**	**Test**
BERTTiny	0.1503	0.1439	0.1526	0.1498	0.1510	0.1483
BERTTiny_h	0.1511	0.1482	0.1505	0.1484	0.1504	0.1459
BERTTiny_ub	0.3762					
DistilRoBERTa	0.5067	0.5186	0.5045	0.5153	0.5046	0.5180
DistilRoBERTa_h	0.5848	0.5978	0.5820	0.5883	0.5832	0.5944
DistilRoBERTa_ub	0.7434					

Table 1: SQuAD F1-score analysis

3 Experimental setup and results

Through the EncoderDecoderModel class of HuggingFace (9), all models pre-trained weights are loaded and leveraged for a subsequent fine-tuning on CoQA. As for the special tokens configuration, the [CLS] token is set as decoder start token and [SEP] token as [EOS] token encoder. Experimental results show that the most effective parameters for beam search decoding are as follows: a maximum and a minimum length for generation equal to 64 and 1 respectively; no repeat ngram of size 3; the length penalty sets to 2; number of beams equal to 15; early stopping sets to *True*. The best training setup instead was found to employ a batch size of 16 and AdamW as optimizer, using a linear scheduler with 10000 warmup steps. All models share the same configuration and training setup, heavily adapted from (3) and (4). Then, each of them is fine-tuned for 3 epochs with 3 different seeds, namely 42, 2022 and 1337, and evaluated using the SQuAD F1-score metric. Obtained results are summarized in Table 1.

4 Discussion

As shown in Table 1, each model obtains comparable results on both the validation and the test sets and the same versions of the same type of model perform similarly across the 3 seeds, according to the SQuAD F1-score. BERTTiny versions do not show significant differences in scores, performing around $0.14/0.15$, while the two versions of DistilRoBERTa obtain different results: the baseline model reaches around $0.50/0.51$, while the history one goes up to $0.58/0.59$. In general, performances seem to be far from great, especially BERTTiny's. However, neither the models versions receiving as input the rationale instead of the passage can reach really high performances, obtaining 0.38 and 0.74 SQuAD F1-scores respectively, as a proof that generative question answering is really an hard task.
To better reason on errors causes, the two best baseline models per type are chosen and the analysis focuses on the 5 hardest conversations per source, in which selected models mostly fail to answer questions. It's interesting to notice that both BERTTiny and DistilRoBERTa models, even when missing the correct answer, in most of the cases predict something at least plausible, as they were able to recognize the question type and coherently answer to interrogative adverbs/pronouns like *Where* or *Who* with an actual place or person respectively. However, only DistilRoBERTa model shows to really predict passage-related answers and it turned out that among all different question types (1), it finds it easier to answer to lexical matches and paraphrasing than pragmatics ones, as the former contain explicit or implicit clues on what portion of the passage the question relates to, while the latter require a common sense knowledge to answer properly. In addition, it emerges that DistilRoBERTa history version performs better than the baseline on questions which refer back to conversational history.

5 Conclusion

The purpose of this work was to address a question answering task through a generative approach, employing two different Seq2Seq models built on BERTTiny and DistilRoBERTa. Obtained results underline that generate free-form answers is not an easy task. While both type of models showed the ability to understand the question type and answer something plausible, even when out of context, only DistilRoBERTa really generated passage-related responses. Future works might focus on the use of larger models, that, even if more computational costly, might lead to better performances.

References

[1] CoQA dataset:
(https://arxiv.org/pdf/1808.07042v2.pdf)

[2] SQuAD F1-score:
(https://docs.allennlp.org/models/main/models/rc/tools/squad/)

[3] BERT2BERT for CNN/Dailymail:
(https://colab.research.google.com/github/patrickvonplaten/notebooks/blob/master/BERT2BERT_for_CNN_Dailymail.ipynb)

[4] Leveraging Pre-trained Checkpoints for Encoder-Decoder Models:
(https://colab.research.google.com/drive/1WIk2bxglElfZewOHboPFNj8H44_VAyKE?usp=sharing#scrollTo=iVKkr2xfNyRh)

[5] Bert-tiny:
(https://huggingface.co/prajjwal1/bert-tiny)

[6] DistiRoBERTa-base:
(https://huggingface.co/distilroberta-base)

[7] BERT:
(https://arxiv.org/pdf/1810.04805.pdf)

[8] RoBERTa:
(https://arxiv.org/pdf/1907.11692.pdf)

[9] HuggingFace:
(https://huggingface.co/)

[10] HuggingFace, Fine-tune a pretrained model:
(https://huggingface.co/docs/transformers/training)