Assignment 2: Question Answering on CoQA dataset: a conversational QA Dataset

NLP Course Project | 3-cfu Project Work | NLP Course Project & Project Work

Nicola Carrassi, Gabriele Colasuonno, and **Antonio Guerra**
Master's Degree in Artificial Intelligence, University of Bologna
{ nicola.carrassi, gabriele.colasuonno, antonio.guerra7 }@studio.unibo.it

Abstract

In this report we will assess the use of *sequence-to-sequence* models to tackle the Question Answering (QA) task on the CoQA dataset. The Question Answering task consists of building systems that, given a context, automatically answer questions in a natural language. In this project in addition to the standard Question Answering we have considered also the Conversational Question Answering which requires the comprehension of both context and historical QAs. Two architectures have been tested: BERT-Tiny and DistilRoBERTa-base. Despite the fact that BERT based models are not designed for generative purposes, with our best configuration (i.e. DistilRoBERTa with QAs History) we achieved 41.46 F1 SQuAD Score.

1 Introduction

There are two main types of QA tasks based on the availability of contextual information: Machine Reading Comprehension (MRC) and Open-domain QA (OpenQA). MRC aims to enable machines to read and comprehend a specified context for answering a given question, instead OpenQA tries to answer without any specified context. Researchers have experimented various machine learning approaches over the years, ranging from traditional algorithms like support vector machines to embedding-based neural approaches like transformers such as (Xiao et al., 2020), with the latter yielding SOTA results in the CoQA dataset (Reddy et al., 2019) which is also the one exploited in this project. In detail it is a dataset for building Conversational Question Answering Systems. The dataset contains 127k questions with answers, obtained from 8k conversations about text passages from seven diverse domains. Different from encoder-only or decoder-only transformer, Natural Language Generation relies on the *sequence-to-sequence* (seq2seq) generation framework which consists of a bidirectional encoder and a unidirectional decoder. In our experiments we have used the seq2seq framework exploiting the same pre-trained models for both the encoder and decoder, the first is Bert-Tiny (Bhargava et al., 2021; Turc et al., 2019) and the second is DistilRoBERTa-base (Sanh et al., 2019). Since in CoQA almost half of questions refer back to conversational history, modeling the dialogue history becomes a critical step towards better understanding the current question. There are several techniques to model the QA history, the most intuitive approach is to prepend the conversation history to the current question (Zhu et al., 2018). In this work we propose a different approach, we have modelled the QAs history by appending it to the context and giving extra tokens embedding information. All our results were obtained by exploiting the Google Colab platform by performing fine tuning for three epochs and evaluating using the F1 SQuAD score.

2 System description

For the implementation and training of the models, we relied on the *transformers* library of Huggingface (Wolf et al., 2020) using Pytorch as backend. We leveraged the EncoderDecoderModel class to build our pre-trained models. Obviously, models can only process numbers, so we used the tokenizers associated with our two models, again provided by Huggingface. For the evaluation of our results we have used the AllenNLP (Gardner et al., 2017) python package that implements SQuAD F1-Score, everything else in the code was produced by us.

3 Experimental setup and results

Since CoQA provides only training and validation sets, we considered the latter as test set and splitted the former at dialogue level to obtain a validation set (80% train set, 20% validation set). Concerning the configuration that takes the QAs history into account, we handled it by appending the QA

pairs up to the current question at the end of the contexts. In addition, we added three special tokens to our tokenizers in order to separate the context from the QA History start and each QA pairs.

Once upon a time, in a barn near... **[HES]** **[HEQ]** What color was Cotton? **[HEA]** white **[HEQ]** Where did she live? **[HEA]** in the barn

---

Figure 1: Example of a Context with QAs History after the special tokens addition.

We trained each of our models for three epochs and repeated each training for the following three seeds: 42, 2022, 1337. In particular, we exploited Huggingface's TrainingAPIs to perform model training; we conducted some tests for hyperparameters and the best results were produced by the following configuration: batch size = 32, weight decay = 0.01, AdamW optimizer, learning rate = 5e-4 for Bert-Tiny and 2e-5 for DistillRoBERTa. Morevoer, we tried to change some parameters in the generation process, in particular we tried to use the beam search instead of the greedy one without getting further improvements. In the following table we illustrate our results.

	F1 SQuAD Score	
Configuration	**Best**	**Mean**
BERT-Tiny	18.78	18.45
BERT-Tiny QAs History	18.74	18.32
DistilRoBERTa	37.57	35.57
DistilRoBERTa QAs History	41.46	38.68

Table 1: Comparison of the different results obtained from each configuration in three different executions.

4 Discussion

As shown in the results table, there is a clear gap between the two models: DistillRoBERTa outperforms BERT-Tiny, with the former achieving an F1 SQuAD Score of 41.46 in the best case and the latter stopping at 18.78. One obvious reason for this difference in results is the different number of parameters of the two models: DistillRoBERTa has approximately 178M parameters while BERT-Tiny only about 9M. This difference may also explain why DistillRoBERTa improves with the addition of QAs history and BERT-Tiny shows no change in results. Moreover, because of the larger number of parameters and the large improvements shown in each training epoch, we have reason to believe that DistillRoBERTa may have improved even more by performing fine tuning for a larger number of epochs. On the other hand, this high number of parameters also has disadvantages, in fact training, evaluation and generation time becomes much longer and consequently consuming much more resources. Anyway, despite the results obtained in our best configuration, we are still far from the SOTA. However, considering that the average number of words in the answers in the CoQA dataset is just three, it is easy that, although the answer is semantically correct, it gets a low F1 SQuAD Score, for example if it uses synonyms. We will now show some examples of generated answers that got zero as F1 SQuAD Score.

Question: For how long?
Real Answer: five-years
Predicted Answer: five years

Question: When did more people start living there?
Real Answer: 1668
Predicted Answer: 2015

Question: Where did she go for her occupation?
Real Answer: Dubai
Predicted Answer: Saudi Arabia

Figure 2: Examples of a generated answers that got F1 SQuAD Score equal to zero.

Looking at the figure above we can see that, even if the answers are not correct, the models managed to understand the context of the questions: it answers with a year when asked for time predictions or with a place when asked for one.

5 Conclusion

In this project we have evaluated the use of generative models for the question answering task on the CoQA dataset, in particular we used two BERT-based architectures as the basis of our encoder-decoder models. As expected our results were very far from the SOTA and this may be due to the fact that BERT based models are not designed for generative purposes. Moreover, as highlighted in the table 8 of CoQA Paper (Reddy et al., 2019), Encoder-Decoder models are the worst at generating answers on this dataset. A straightforward way to achieve better results could be the use of specifically designed models for generative purposes like BART (Lewis et al., 2020) or T5 (Raffel et al., 2020).

References

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Dongling Xiao, Han Zhang, Yu-Kun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE-GEN: an enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3997–4003. ijcai.org.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. Sdnet: Contextualized attention-based deep network for conversational question answering. *CoRR*, abs/1812.03593.