Assignment 2

Davide Femia, Riccardo Paolini, Sfarzo El Husseini and **Alessandro D'Amico**
Master's Degree in Artificial Intelligence, University of Bologna
{ davide.femia, riccardo.paolini5, sfarzo.elhusseini, alessandro.damico5 }@studio.unibo.it

Abstract

In this paper we address the problem of Conversational Question Answering, with reference to the CoQA dataset introduced by Stanford University in (Reddy et al., 2018). One of the major challenges of this tasks is the management of the conversation history to answer the current question. We propose a solution based on transformers such as BERTTiny and DistilRoBERTa. In particular, our architectures consist of two networks, the first deals with the rationale extraction (the sentence which contains the answer), while the latter identifies the correct answer and reformulates it. This pipeline should simplify the task for both networks.

1 Introduction

Conversational question answering is a task that requires the ability to correctly interpret a question in the context of previous conversation turns. The response to be given is free-form and should appear to be given by a human (Egonmwan and Chali, 2019).
Previous works are based on two main approaches:

- Encoder-only architectures that aim at extracting the answer directly from the passage.
- Encoder-Decoder architectures in which the decoder attempts to generate a response dependent on the input provided by the encoder.

Qu et al. showed how effective an encoder-only architecture that responds by imitating the text can be. On the other hand, encoder-decoder models have been proved to be really effective for causal language modelling thanks to their ability to generate free-form text conditioned on the context. Our approach aims at combining the advantages of both architectures. In particular, we used the two models by stacking them together, so that the encoder-only model filters the information that will be passed to the encoder-decoder (Yu and Liu, 2021).

2 System description

Our implementation is based on two networks, as shown in Figure 1 and Figure 2. The first one is an encoder-only model (**Span Extractor**) which precisely extracts the salient sentences (rationale) of the passage, depending on the given question and the history of the conversation (Devlin et al., 2018). The second one is an encoder-decoder model (**Answer Generator**) that identifies the answer lying in the rationale and refines it.

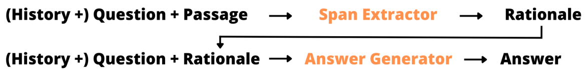
(History +) Question + Passage
Span Extractor
Rationale
(History +) Question + Rationale
Answer Generator
Answer

Figure 1: Data Flow

In particular, we created new model classes for our BERTTiny-based / DistillRoBERTa-based models.BERTiny and DistilRoBERTa are both compact versions of their respective models, BERT and RoBERTa. BERTiny focuses on reducing model size and sacrificing performance, while DistilRoBERTa prioritizes model compression while retaining most of the original model's performance. Each of these classes contains its own forward and generate methods, which in turn recall the forward / generate method of the underlying models, but they also include the necessary code to produce a suitable input for the answer generator by using the prediction made by the span extractor.

3 Experimental setup and results

The most popular dataset concerning this task is CoQA, which contains 8k conversations of different lengths. The peculiarity of this dataset is the naturalness of the answers. The CoQA dataset directly provides training set and test set. The training set was splitted into training and validation set, taking 80% and 20% of the dialogues respectively. We trained each model using three different seeds, {42, 2022, 1337}, and then we averaged the results.

Accordingly with transformers' requirements we pass two tokenized sequences to the model, the question (plus history) and the passage. During the tokenization step, we do not preprocess the text since transformers tokenizers are already pretrained on huge amounts of raw data.

Moreover, we use inputs of 512 tokens, so we padded/truncated sequences accordingly. As regards history management, the history of the conversation is prepended to the current question instead of creating a special embedding. The hyper-parameters are reported in Table 1.

	BS	**Optim**	**WD**	**LR**	**H**
B	32	AdamW	0.01	5e-5	0
B (H)	32	AdamW	0.01	5e-5	4
R	4	AdamW	0.01	5e-5	0
R (H)	4	AdamW	0.01	5e-5	4

Table 1: Hyper-parameters. ***B***: BERTTiny; ***R***: DistilRoBERTa; **BS**: batch size; **WD**: weigth decay; **LR**: learning rate; **H**: history length. The optimizer uses default betas and epsilon.

For all the models we used a teacher forcing approach to ease the training of the answer generator in the early steps, since the span extractor initially provides imprecise rationales. The teacher forcing probability decays linearly from 1.0 to 0.3 during training.

The results obtained on CoQA dataset are summarized in Tables 2a and 2b.

4 Discussion

The first thing that can be noticed by watching the results is that DistillRoBERTa is clearly overperforming BERTTiny. The huge gap is explained by the different capacity of the two models, where the first contains 82 million parameters compared to only 4.4 million in the other.

In particular, larger models have been found to be much better at answering 'WH questions' that are those questions that contain 'what', 'when', 'where', 'which', 'who', 'how', 'whose' or 'why'. These questions are not so easy to be answered since they do not require a binary answer and are usually context-specific.

Another thing that is interesting to observe is the strong bias BERTTiny has in predicting YES rather than NO, this fact is well highlighted by the performance discrepancy in YES/NO answers. Contrarily, DistillRoBERTa does not suffer from this problem having even a slight preference to answer NO. During error analysis, we found that when the span extractor fails to predict the rationale then the answer is completely wrong, that's due to the fact that the answer generator cannot see the relevant part of the passage to provide a good response. Another particular fact is that when the question is about a well-known thing such as a famous person or event, the model, even if it is wrong, tends to predict something relevant in that context.

Q: 'What is Mayweathers nick name?'
GT: 'is the money man'
A: 'diego pacquiao' / 'tom pacquiao' / 'nick'

where Diego Pacquiao and Tom Pacquiao are combinations of famous boxers: Manny Pacquiao and Diego Corrales.

In addition, the models that do not consider the history can't answer history dependent answers like the following.

Q: 'and what is his daughter's name?'
GT: 'Miss Harding'
A: 'Barbara' / 'Barbara' / 'Mallory'

Here, as in most of the cases, it predicts something which is reasonable, in fact, it answers with female names however it does not provide the right answer.

5 Conclusion

In our project, we proposed models that mainly introduce changes in the network architecture. We focused on separating the tasks of rationale extraction and answer generation, thus making prediction easier for both networks. These models were then used for question answering on the CoQA dataset. The results showed that our BERT-Tiny model achieves performances in line with standard Seq2Seq models (Reddy et al., 2018), while DistillRoBERTa improves the overall SQUAD F1-Score of about 25%, proving to be quite good even in answers that require rephrasing the text. However, its computational cost was considerably higher than BERT-Tiny, requiring three times more training time.

Many other techniques can be used to further improve results, such as sliding windows that divide the passage into smaller sequences, or a fully differentiable span extractor that can also benefit from training the answer generator following it, or also fancier ways of modeling the history like the history answer embedding used by Qu et al.

6 Links to external resources

Link to the GitHub repository Question-Answering with Transformers.

7 Auxiliary material

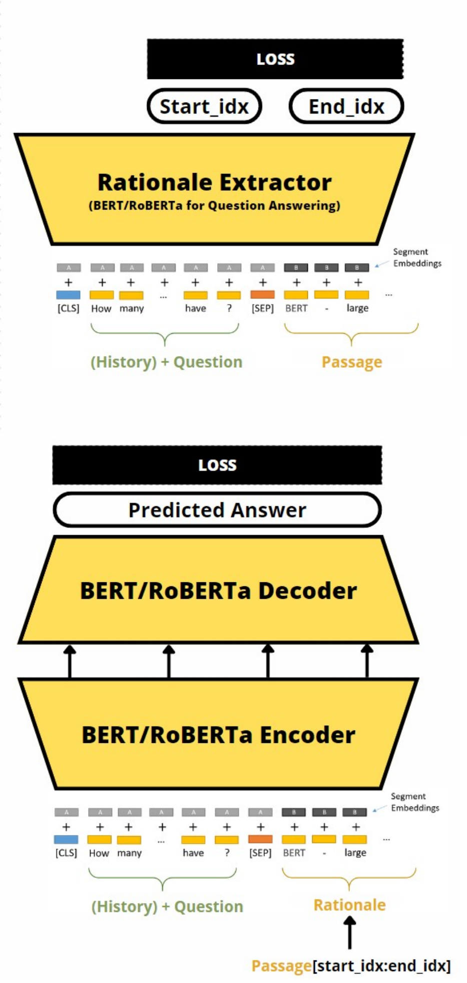

Figure 2: Model architecture: on the left we have the encoder for rationale extraction, on the right the encoder-decoder model that generates the answer using the rationale instead of the whole passage.

(a) BERTTiny Performance Averaged Across Seeds

Category	Percentage	Avg F1-score
Avg F1-score		0.2179
YES	11.2%	0.8394
NO	9.7%	0.1988
WH- questions	73.9%	0.1412
Multiple choice	1.0%	0.1739

(b) DistilRoBERTa Performance Averaged Across Seeds

Category	Percentage	Avg F1-score
Avg F1-score		0.4923
YES	11.2%	0.6894
NO	9.7%	0.7459
WH- questions	73.9%	0.4495
Multiple choice	1.0%	0.3330

Table 2: SQUAD F1-score measures the average overlap between the prediction and ground truth answer. The prediction and ground truth are treated as bags of tokens, and F1 is calculated on them (Rajpurkar et al., 2016).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Elozino Egonmwan and Yllias Chali. 2019. Transformer and seq2seq model for paraphrase generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 249–255, Hong Kong. Association for Computational Linguistics.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019a. BERT with history answer embedding for conversational question answering. *CoRR*, abs/1905.05412.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019b. BERT with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042.

Puning Yu and Yunyi Liu. 2021. Roberta-based encoder-decoder model for question answering system. In *2021 International Conference on Intelligent Computing, Automation and Applications (ICAA)*, pages 344–349.