Assignment 2

Substitute the ↑ title ↑ with your project's title, or with Assignment 1 / 2

↓ Keep only one of the following three labels / leave empty for assignments: ↓

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

DO NOT MODIFY THIS TEMPLATE - EXCEPT, OF COURSE FOR TITLE, SUBTITLE AND AUTHORS.
IN THE FINAL VERSION, IN THE LaTeX SOURCE REMOVE THE `guidelines` OPTION FROM `\usepackage[guidelines]{nlpreport}`.

NOTICE: THIS REPORT'S LENGTH MUST RESPECT THE FOLLOWING PAGE LIMITS:

- ASSIGNMENT: **2 PAGES**
- NLP PROJECT OR PROJECT WORK: **8 PAGES**
- COMBINED NLP PROJECT + PW: **12 PAGES**

PLUS LINKS, REFERENCES AND APPENDICES.
THIS MEANS THAT YOU CANNOT FILL ALL SECTIONS TO MAXIMUM LENGTH. IT ALSO MEANS THAT, QUITE POSSIBLY, YOU WILL HAVE TO LEAVE OUT OF THE REPORT PART OF THE WORK YOU HAVE DONE OR OBSERVATIONS YOU HAVE. THIS IS NORMAL: THE REPORT SHOULD EMPHASIZE WHAT IS MOST SIGNIFICANT, NOTEWORTHY, AND REFER TO THE NOTEBOOK FOR ANYTHING ELSE.
FOR ANY OTHER ASPECT OF YOUR WORK THAT YOU WOULD LIKE TO EMPHASIZE BUT CANNOT EXPLAIN HERE FOR LACK OF SPACE, FEEL FREE TO ADD COMMENTS IN THE NOTEBOOK.
INTERESTING TEXT EXAMPLES THAT EXCEED THE MAXIMUM LENGTH OF THE REPORT CAN BE PLACED IN A DEDICATED APPENDIX AFTER THE REFERENCES.

Abstract

The abstract is very brief summary of your report. Try to keep it no longer than 15-20 lines at most. Write your objective, your approach, and your main observations (what are the findings that make this report worthwhile reading?)

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

MAX 1 COLUMN FOR ASSIGNMENT REPORTS / 2 COLUMNS FOR PROJECT OR PW / 3 FOR COMBINED REPORTS.

Then give a short overview of known/standard/possible approaches to that problems, if any, and what are their advantages/limitations. Conversational question answering is a task that requires the ability to correctly interpret a question in the context of previous conversation turns. 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 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., 2019) 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. After that, discuss your approach, and motivate why you follow that approach. If you are drawing inspiration from an existing model, study, paper, textbook example, challenge, . . . , be sure to add all the necessary references (Chowdhery et al., 2022; Lorenzo et al., 2022; Antici et al., 2021; Nakov et al., 2021; Röttger et al., 2022; Lippi and Torroni, 2016).[1] 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.

Next, give a brief summary of your experimental setup: how many experiments did you run on which dataset. Last, make a list of the main results or take-home lessons from your work.

HERE AND EVERYWHERE ELSE: ALWAYS KEEP IN MIND THAT, CRUCIALLY, WHATEVER TEXT/CODE/-FIGURES/IDEAS/... YOU TAKE FROM ELSEWHERE MUST BE CLEARLY IDENTIFIED AND PROPERLY REFERENCED IN THE REPORT.

2 System description

MAX 1 COLUMN FOR ASSIGNMENT REPORTS / 4 COLUMNS FOR PROJECT OR PW / 6 FOR COMBINED REPORTS.

Describe the system or systems you have implemented (architectures, pipelines, etc), and used to run your experiments. If you reuse parts of code written by others, be sure to make very clear your original contribution in terms of

- architecture: is the architecture your design or did you take it from somewhere else
- coding: which parts of code are original or heavily adapted? adapted from existing sources? taken from external sources with minimal adaptations?

It is a good idea to add figures to illustrate your pipeline and/or architecture(s) (see Figure 1)

Our implementation is based on two networks, as shown in Figure 2 and Figure 3. 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.

In particular, we created new model classes for our BERTTiny-based / DistillRoBERTa-based models. 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

MAX 1 COLUMN FOR ASSIGNMENT REPORTS / 3 COLUMNS FOR PROJECT OR PW / 5 FOR COMBINED REPORTS.

Describe how you set up your experiments: which architectures/configurations you used, which hyper-parameters and what methods used to set them, which optimizers, metrics, etc.
Then, **use tables** to summarize your your findings (numerical results) in validation and test. If you don't have experience with tables in LaTeX, you might want to use LaTeXtable generator to quickly create a table template. 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 are already pre-trained 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.
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

[1] Add only what is relevant.

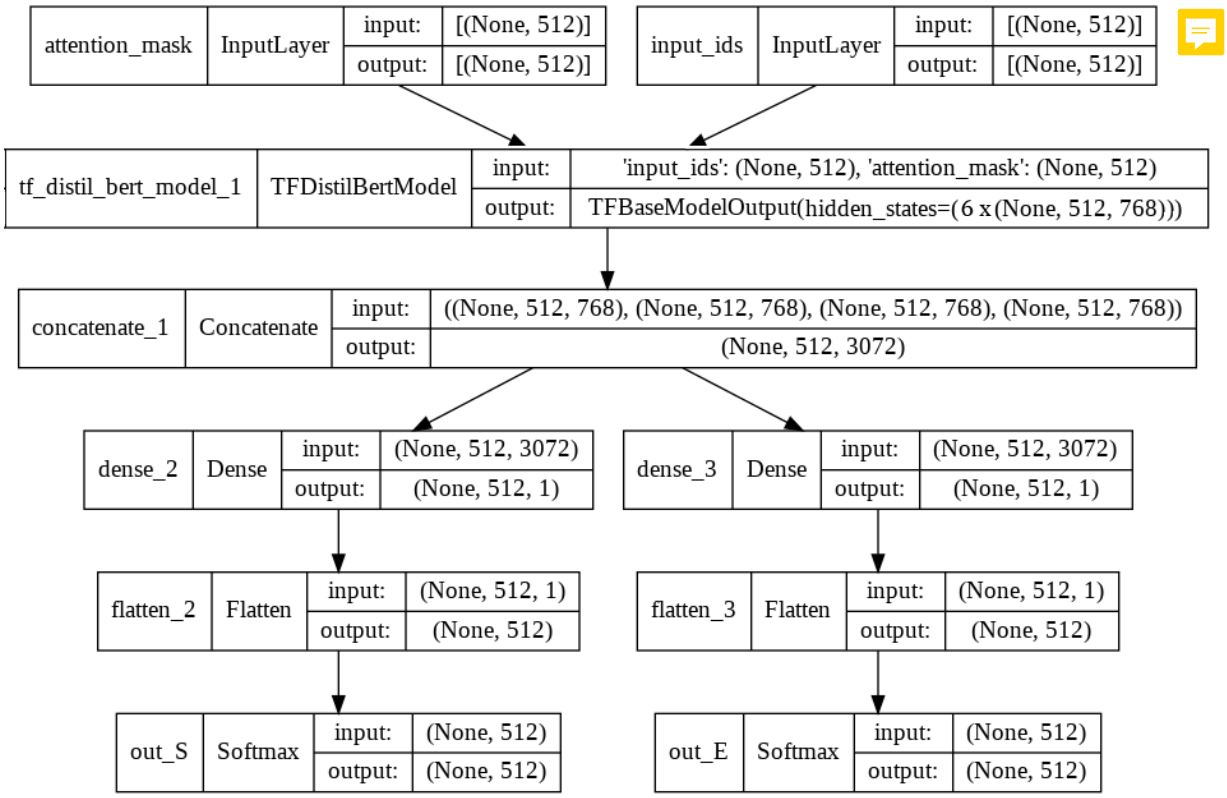

Figure 1: Model architecture

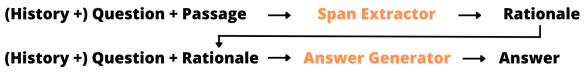

Figure 2: Data Flow

	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.

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 the clear superiority of DistillRoBERTa-based models over BERTTiny-based models. 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.

MAX 1.5 COLUMNS FOR ASSIGNMENT REPORTS / 3 COLUMNS FOR PROJECT / 4 FOR COMBINED REPORTS. ADDITIONAL EXAMPLES COULD BE PLACED IN AN APPENDIX AFTER THE REFERENCES IF THEY DO NOT FIT HERE.

Here you should make your analysis of the results you obtained in your experiments. Your discussion should be structured in two parts:

- discussion of quantitative results (based on the metrics you have identified earlier; compare with baselines);
- error analysis: show some examples of odd-/wrong/unwanted outputs; reason about why you are getting those results, elaborate on what could/should be changed in future developments of this work.

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, fully differentiable span extractor that can also benefit from training the answer generator following it, or also fancier ways of modeling the history

MAX 1 COLUMN.

In one or two paragraphs, recap your work and main results. What did you observe? Did all go according to expectations? Was there anything surprising or worthwhile mentioning? After that, discuss the main limitations of the solution you have implemented, and indicate promising directions for future improvement.

6 Links to external resources

THIS SECTION IS OPTIONAL

- a link to your GitHub or any other public repo where one can find your code (only if you did not submit your code on Virtuale);
- a link to your dataset (only for non-standard projects or project works).

DO NOT INSERT CODE IN THIS REPORT

(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

References

Francesco Antici, Luca Bolognini, Matteo Antonio Inajetovic, Bogdan Ivasiuk, Andrea Galassi, and Federico Ruggeri. 2021. SubjectivITA: An Italian corpus for subjectivity detection in newspapers. In *CLEF*, volume 12880 of *Lecture Notes in Computer Science*, pages 40–52. Springer.

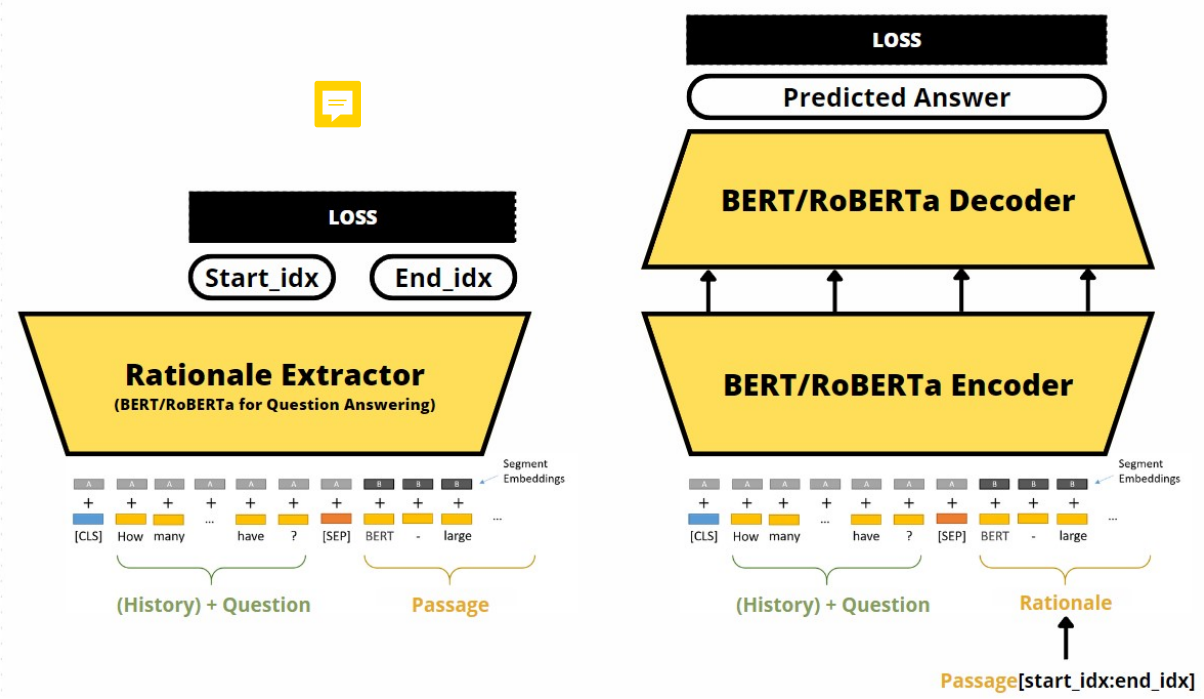

Figure 3: Encoder-Decoder Architecture

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

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.

Marco Lippi and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology*, 16(2):10:1–10:25.

Abelardo Carlos Martinez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-semantic parsing and generation: the BabelNet meaning representation. In *ACL (1)*, pages 1727–1741. Association for Computational Linguistics.

Preslav Nakov, David P. A. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated fact-checking for assisting human fact-checkers. In *IJCAI*, pages 4551–4558. ijcai.org.

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

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

Paul Röttger, Bertie Vidgen, Dirk Hovy, and Janet B. Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective NLP tasks. In *NAACL-HLT*, pages 175–190. Association for Computational Linguistics.