Assignment 2
Question Answering (QA) on CoQA dataset: a conversational QA dataset.

Vincenzo Collura, Gianmarco Pappacoda, and **Anthea Silvia Sasdelli**
Master's Degree in Artificial Intelligence, University of Bologna
{ vincenzo.collura2, gianmarco.pappacoda, anthea.sasdelli }@studio.unibo.it

Abstract

Question Answering (QA) is a relevant task of Natural Language Processing, albeit simple it comes in many different forms and evaluating text comprehension is a matter of debate. The objective of this report is to explore the use of two Large Language Models (LLM) used in an encoder-decoder fashion to perform QA on the CoQA. The task is considered in two variants: with and without considering dialogue history.

1 Introduction

Question Answering systems are becoming increasingly necessary as many tasks can benefit from a prompt reply to a given question. Most Question Answering systems work in the following way: given a passage (often called context) and a question, they try to extract a span of text that answers the question. In the given task, the models should generate an answer rather than extracting it from the context. Two variants of the task have been considered:

- **Question+Context:** as described above.
- **Question+Context+History:** the question/-context pair is enriched with previous passages in the dialogue.

The dataset given for this assignment is the Conversational Question Answering Challenge (CoQA) (Reddy et al., 2018). As in the dataset a number of unanswerable QA pairs is present (1.3%), they have been removed as per instructions.

In order to evaluate the impact of the history, two models have been created:

- $A = f_\theta(Q, P)$
- $A = f_\theta(Q, P, H)$

Where Q is the question, P is the text passage (context), H is the dialogue history, A is the generated answer and f_θ is the transformer-based model with θ parameters.

2 System description

The system can be synthesized as follows:

- Preprocessing
- Model instantiation
- Tokenization
- Training
- Answer generation (Inference)

The above steps have been implemented through a wrapper which abstracts the main function expected of these models from the underlying interface (HuggingFace transformers/datasets libraries).

The data preprocessing has been kept to a minimum as per instructions, the only preprocessing step being the removal of unanswerable QA pairs. As for history, each previous QA pair is encoded as text and is appended to the context.

The proposed models have been implemented using an Encoder-Decoder architecture, where Large Language Models (LLMs) are used to encode a given input and produce a given output. In the task at hand, this architecture is leveraged to repurpose LLMs for task they have not been trained for, a form of transfer learning.

The tokenizer, the component responsible for tokenization, is wrapped alongside the underlying model as it is dependent on it.

The training step is once again a wrap of Hugging Face Seq2SeqTrainer.

The inference step can be performed in many different ways as the answer is generated rather

than extracted. Three main methods have been implemented: **Beam Search**, **Top-K** and **Greedy Search**.

As LLMs impose a hard constraint on resources needed for training, the following two "distilled" versions of popular LLMs have been employed:

- `BERT-Tiny` (Bhargava et al., 2021; Turc et al., 2019) This is one of the smaller pre-trained BERT variants..
- `DistilRoBERTa-base` (Sanh et al., 2019)

As even the two "distilled" versions of the proposed LLMs proved to be heavy for the authors' resources, the weights of the encoder and decoder parts have been tied as it has been empirically proven to only slightly reduce performance (Rothe et al., 2020).

3 Experimental setup and results

The data was split into training and validation sets, with a 80% ratio. The test set is already provided. During each experiment the Random Number Generator (RNG) of all the involved frameworks has been fixed to ensure reproducibility. The experiments, both training and evaluation, are repeated using three different seeds.

The most important hyperparameters available for tweaking are: learning rate, batch size and text generation method. Bert-tiny models have been trained using $lr = 10^{-3}$, distilroberta-base ones with $lr = 2 \cdot 10^{-5}$.

During the evaluation step, the models are evaluated against the validation and test set performing text generation using **Beam search**. The text generation method is tweak-able, Beam search has been observed to empirically produce the best results.

The metric selected for evaluation is the SQUAD F1-score(Rajpurkar et al., 2018) (implementation thanks to (Gardner et al., 2017)).

		Bert-Tiny		DistilRoBERTa-base	
History	seed	squad f1-score			
		val	test	val	test
No	42	0.1639	0.1638	0.1728	0.1770
	2022	**0.1705**	**0.1700**	0.1752	0.1712
	1337	0.1701	0.1645	**0.1870**	**0.1873**
Yes	42	0.1700	**0.1710**	0.1732	0.1759
	2022	0.1702	0.1699	0.1780	0.1704
	1337	**0.1738**	0.1646	**0.2074**	**0.2033**

4 Discussion

While it has been possible to implement the system and the experiments, the resulting models have been unable to provide satisfactory results. Most of the answers generated by the models are wrong, albeit they show a striking resemblance with an answer that makes sense: the models produce answers that are in the domain of the correct answer.

The addendum of the history has not yielded significantly better models results compared to the baseline versions that did not feature history as an input. This is partly due to the fact the history is appended to the context, the models maximum input length limits its effectiveness.

The obtained results are lacking with respect to the selected metric but show interesting patterns in the learning process of the models. As most answers are considered wrong, it is not possible to assess common errors. It is, instead, feasible to observe that most correct answers are closed-form yes/no answers.

5 Conclusion

It has been possible to implement all the given tasks. The models employing history have been empirically proven to be slightly superior with respect to their non-history variants. While it has been possible to obtain some results, the limit on the number of epochs and the lack of required equipment to train the models have severely limited the possibility to obtain significative results. As a result of this the error analysis did not highlight anything in particular as both models are incapable of answering to most questions and most of the positive results are due to closed yes/no questions.

Moreover, the method used to implement history (i.e. concatenating history and context) has proven to be somewhat useful, but it is once again limited by the input length which is in turn limited by the available resources to train the model. A different approach would have required different architectures and/or multiple networks, which was again, not allowed.

Overall distilroberta-base performed only slightly better compared to bert-tiny (w.r.t squad-f1) while using many more parameters, resources.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. Technical report. ArXiv:1806.03822 [cs] type: article.

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

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280. ArXiv:1907.12461 [cs].

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.