Assignment 2

Alessandro Lombardini, Giacomo Melacini, Matteo Rossi Reich and **Lorenzo Tribuiani**
Master's Degree in Artificial Intelligence, University of Bologna
{alessandr.lombardin3, giacomo.melacini, matteo.rossireich, lorenzo.tribuiani}@studio.unibo.it

Abstract

Generative Question Answering (QA) is a particular task of NLP which aims to creates models and architectures capable of producing answers to given questions in the form of free text. The aim of this assignment is to create a QA system based on the *CoQa* dataset by fine-tuning the *distilroberta-base* and *bertiny* models from *Huggingface*

1 Introduction

The development of a *Question Answering System* has been widely based on the *Huggingface's Transformers* **Model for Question Answering** and **Encoder Decoder Model**. The fist model is mainly used for *Extractive* question answering (HuggingFace, b), which doesn't match completely the task requested since a *Generative* QA is required. The second one is generally used for *Sequence generation* (HuggingFace, a). In order to develop a *generative question answering* the two models have been combined together: the first model is trained to extract the *rationale* from the context given the question and the context itself while the second model is trained to produce the wanted answer from the produced rationale and again the question. The complete model has been trained and evaluated on the *CoQa* dataset with different results:

- *distilroberta-base* based model has reached good results with a f1 score of 0.60 and good generalization
- *bertiny* based model had the general worst results

2 System description

As previously introduced the main architecture is based on two models. The *Model For Question Answering* of HuggingFace offers a base model (which in this case would be *distilroberta-base* or *bertiny*) with a *question answering* head, which is basically composed by dense layers for the classification of the two start and end indices of rationale. The *Encoder Decoder Model* is a *sequence to sequence* model that use two general models, one for the encoding of the inputs and one for the decoding of the outputs.
Since those models have a limited input dimension, for the span extractor model a sliding window has been developed. This part of the code is widely a readjustment of the code reported from the Huggingface's documentation itself(HuggingFace, b).
The **complete model** is basically made up by the concatenation of the two previous models: The inputs (context and question) are passed into the *span extractor model*. This model will produce as output the expected rationale which will be passed, together with the question, into the *encoder decoder model* producing the final answer for the given question.

3 Data

Data preprocessing is a fundamental step for the development of this assignment. The inputs of the *span extractor* model have been regularized with *padding*[1] and *truncation* to the same size. In order to keep track of the overflowing tokens the context is divided in chunks (in case of truncation) and each input is made up as $(question + chunk_i)$.
The labels are updated according to this division with $(0, 0)$ if the rationale is not in the current chunk of the context, $(span_{start}, span_{end})$ elsewhere. Rationales divided into more than one chunk are ignored for simplicity.
An history field is eventually created by reporting the couples (question, answer) of the $n-1$ previous questions for auto-referring answers and inserted into the context.

[1] specific series of token added to the sequence in order to match the wanted input size

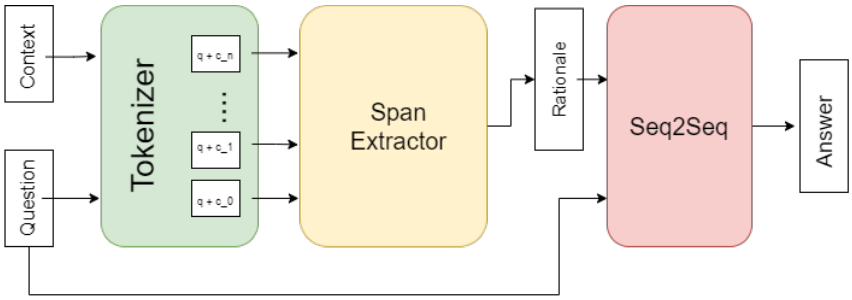

Figure 1: Model architecture

4 Experimental setup and results

Each model has been trained on the *CoQa* train and validation datasets with three different seeds (42, 2022, 1337) both with and without history. The following table shows the mean f1 score over the test set for each model:

	mean f1 score		
	seed	without history	with history
roberta	42	0.588	0.457
	2022	0.584	0.459
	1337	0.585	0.459
bertiny	42	0.077	0.080
	2022	0.077	0.092
	1337	0.076	0.092

Table 1: F1 score over the test set

In addition to the previos analysis a mean evaluation of the single models (*Span Extractor* and *Sequence to Sequence*) is proposed.

		Mean f1 score	
		no history	history
roberta	span ex.	0.48	0.35
	seq2seq	0.72	/
bertiny	span ex.	0.008	0.009
	seq2seq	0.132	/

Table 2: Mean f1 score of the single models on the evaluation set

5 Discussion

The overall data acquired during the training and validation of the models shows, in first place, a strong difference between the way the two models exploit the results. As shown in table 1, *distilroberta-base* seems to be much more accurate than *bertiny* both with and without history, probably due to the big difference in size of the two models. In the second place, table 2 shows how the *span extractor* model performs much porely compared to the *sequence to sequence* model which leads to a decreased accuracy on the overall results. The error analysis over the 5 worst result for each model has brought the following conclusions:

- all the models shows difficulties in discriminating the answers *yes* and *no*, the network has efficiently learned when to reformulate the answer into one of the two, but still makes confusion over them.
- some of the answers are marked as uncorrect due to the peculiar *squad f1* used. Answers like *The Farmer* and *The Farmer's* has a 0 score since the articles are not taken in account and and the two words are marked as completely different

Surely, increasing the performance of the *span ex* model would be a huge step forward. Inserting the history inside the question would grant it to be present in all the chunks and not only in the first one leading to a possible increase of results.

6 Conclusion

Despite the expectation the complete model has shown good results on a task relatively complex. A future step could be exploring different model, pre-trained on the specific task requested by each inner model (span extraction and text generation) in order to have highly expert system on each end of the model for more accurate results.

References

HuggingFace. a. Encoder decoder models.

HuggingFace. b. Question answering.