Q&A without context
Machine Learning for Natural Language Processing 2021

Sofiane Ettayeb
3A ENSAE Voie DSSA
sofiane.ettayeb@ensae.fr

Romain Ilbert
3A ENSAE Voie DSSA
romain.ilbert@ensae.fr

Abstract

Our project focuses on the Question Answering (QA) task. For this we trained an extractive model on the SQUAD V2 dataset: https://rajpurkar.github.io/SQuAD-explorer/. Then, we used a parser to get contexts from wikipedia to answer questions without context input.

1 Problem Framing

The goal of this project is to build a simple rudimentary chatbot able to answer general knowledge questions. In order to do that we will first study the task of extractive question answering. Given a question and a context, can we find the answer to the question in the context ? How can we get an answer when there is no context?

2 Experiments Protocol

For the task of extractive question answering, we decided to use a variation of BERT (Chang M.-W. Lee K. Devlin and Toutanova, 2018) called DistilBERT (Sanh et al., 2019) to obtain our embeddings. This version of BERT is lighter and faster while not losing too much of BERT performances (95% of BERT's performances as measured on the GLUE language understanding benchmark) which suits our ambitions on this project. We don't aim to obtain state of the art results but to better understand the strengths and weaknesses of such a model. In general, BERT language model has yielded very good result on the SQUAD Dataset, some model using variations of bert (Lan et al., 2019) yield better than human results on this task.

As BERT handles a separator token, we will concatenate the question and the context as a single padded or truncated input (such that all of our inputs have the same size) and feed it into the model. We use the last hidden layer $X \in \mathbb{R}^{512*768}$ as our input representation. Each of the 512 tokens of our input is represented by a vector $T_i \in \mathbb{R}^{768}$

Extracting the answer in the context is equivalent to predicting the position of the first and the last tokens of the answer. For each token in our input, we want the probability of it being the first and the last token of the answer (the context of containing the answer is in the input). Thus we add a linear layer $L : \mathbb{R}^{768} \longrightarrow \mathbb{R}^2$ to our model which is applied to each token T_i. We can then apply the softmax function over every tokens to represent the desired probability distributions S and E both in $\mathbb{R}^{512}$. In the original BERT paper, the predicted models are $\arg\max_{j \geq i} E_i + S_j$ to ensure the predicted end is after the predicted start. To simplify our computations, we will first compute $i_{pred} = \arg\max_i E_i$ and $j_{pred} = \arg\max_{j \geq i} S_j$.

To answer questions without context, we decided to use a Wikipedia parser. Given a question as a query, it would consult the top 5 pages related to the query words, parse them and separate them into different chunks of text that we will feed into our previous model as the context. It will then return the 5 answers with the highest confidence score (defined as the sum of the start score and end score). To improve our model, we want it to be able to predict when a context doesn't have the answer to the question as many bad contexts will be fed into it. For that we will train our model on the version 2 of SQUAD which include unanswerable questions. When the question has no answer, our model will point toward the index 0 corresponding to the special [CLS] token. We will evaluate our model using the f1 score and Exact matching score. The score is calculated on the answer splited into words using an adaptation of SQUAD official evaluation script. We also define a threshold parameter τ such that our model will predict "no answer" if $L * T_{i_{pred},0} + L * T_{j_{pred},1} < \tau + L * T_{0,0} + L * T_{0,1}$ where i_{pred} and j_{pred} are

defined in the previous paragraph and $*$ is the dot product. Finaly, we evaluated the importance of the format of the input. In BERT official documentation, they put the question first and then the context. In huggingface popular implementations, they did they reverse. We'll test both implementations which makes a total of 4 different models. 2 models have been trained on only answerable questions with a batch size of 32 for 3 epochs. The other 2 have been trained on every questions with a batch size of 48 for 2 epochs. In both case we used cross entropy loss.

3 Results

3.1 Quantitative evaluation on question answering with context

We evaluated the F1 score and Exact matching score on the validation dataset. We separated the score on questions with answer and questions without answer. On unanswerable question, the F1 score is equal to the EM score. in the following table, CQa stands for "Input context then question trained on all answers" while "n" indicate that it has only be trained on questions with answers

Model	F1a	EMa	EMn	F1	EM
CQa	74	66	61	68	64
QCa	70	64	65	68	65
CQn				83	74
QCn				84	75

The model trained to answer only question yield good results when evaluated (on squad v1 we would have been ranked 40th). Comparatively, the models trained on question with and without answers yield worse result but we have yet to take into account the thresholding. An interesting thing to note is that inputing the context first allows the model to perform better on questions with answers while inputing the question first allows the model to better recognize question without answers according to the scores.

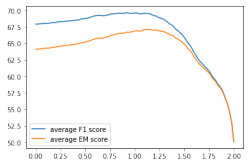

Figure 1: Evolution of the scores of CQa with τ

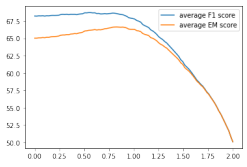

Figure 2: Evolution of the scores of QCa with τ

With a threshold of 0.8, we increase the scores of both models by 1-2

3.2 Qualitative evaluation on question answering with context

We mainly observed 4 kinds of question/context where our models had trouble answering correctly the question:

1. Trap question which often plays on contradictions (for instance asking about the name of the husband when only the wife is mentioned). These questions are often unanswerable with the given context
2. Questions where the answer is indirectly mentioned in the context
3. Bad questions which where still present in the dataset. The answer was often debatable or didn't make much sense. Often reformulating the question allowed the model to find better answers

3.3 Evaluation on question answering without context

We selected the top 5 answers with highest confidence. Evaluating quantitatively is complicated as the model execution is lengthy enough in time. Qualitatively. The model performed quite good on definition-type questions (For instance "What does NLP stand for ?"). During our testing, we found out that our model has the most difficulties answering question needing comparisons (For instance when asked about the tallest building in the world, it answered the tallest building in a specific Country) and question asking for dates. The model trained with unanswerable questions performed overall better than the other 2 and the model QCa performed slightly better than its counterpart CQa.

4 Discussion/Conclusion

The model performed better than we expected considering its simplicity. It can be attributed mainly to the power of transformer-like model such as BERT. One big issue was finding the correct wikipedia page according to the query. One possible way to improve our algorithm would be to add a keyword extractor using NER on the questions to search for the relevant pages. Another way would be to use heavier and better BERT like model for the question answering part.We could also endeavour to further fine tune our model on other QA datasets like TriviaQA

.

References

J. Chang M.-W. Lee K. Devlin and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional trans- formers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

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