SQUAD Question-Answering With Transformers
Machine Learning for Natural Language Processing 2020

Christos Katsoulakis
Ensae Paris
christos.katsoulakis@ensae.fr

Eric Lavergne
Ensae Paris
eric.lavergne@ensae.fr

Abstract

In this report, we briefly present our work for the final evaluation of the Machine Learning for NLP course. We chose to focus on the extractive question-answering task on SQUAD 1 and 2. We compared training a transformer from scratch and fine-tuning a BERT-based model pre-trained on language modelling. The python code is available in Google Colab [1] and in GitHub [2].

1 Problem Framing

1.1 Question-Answering

In a context of abundance of textual documents, finding relevant information can be complicated. Developing tools to correctly answer a question given a document would be very useful to gain efficiency. This is why we address one of the problems of Question Answering that consists in extracting the answer to a question given a document. More generally, Question Answering is becoming more and more interesting to provide a solution to many practical needs to extract relevant information and advanced methods are used in many applications that are massively adopted today, such as virtual assistants or internet research.

1.2 Answerability detection

When searching for information in a large volume of documents, the answer to a question will not always be contained in the document browsed. In addition to correctly answering a question when the document contains the information, it is important that the model be able to identify whether the question is answered or not in a document. If there is no answer, the model should indicate it rather than trying to predict an incorrect answer.

[1] Colab file
[2] Github repository

1.3 SQUAD datasets insights

To carry out this project the Stanford Question Answering Dataset (SQuAD) has been used. This dataset contains many examples that are varied, which is interesting to produce a model that can handle a wide variety of contexts and types of questions. Moreover, there are two versions of this dataset: SQuAD 1 which only has questions whose context admits an answer (Pranav Rajpurkar, 2016), SQuAD 2 which additionally incorporates questions without answers in the context (Pranav Rajpurkar, 2018). Thus a problem of increasing difficulty can be addressed. The training datasets have respectively 87,599 and 130,319 questions on a set of Wikipedia articles.

2 Experiments Protocol

2.1 Transformer with Glove

We first trained a model from scratch based on the Transformer architecture.

The Transformer architecture is a substitute for Recurrent Neural Networks whose most popular entities are the LSTM and the GRU. In a Transformer, recurrence is replaced by the attention mechanism; data are no longer processed sequentially but in parallel. We chose to train a small-scale transformer, with 4 blocks comprising 4 heads each and feedforward networks with 256 hidden neurons.

The only pre-trained elements of the model correspond to the embeddings, for which we used Glove vectors of size 100. We selected them in particular because they were built with data from the same domain as SQuAD 1 and 2 (Wikipedia).

In addition there are positional embeddings that encode the position of each element of the input sequence. They are essential since the model processes the data sequentially independently of the order of the elements in the sequence. We have

chosen to apply sinusoidal embeddings.

For the output, a feedforward network with two output neurons is used to assign to each element of the sequence the probability of being the first or the last token of the response to be extracted.

2.2 BERT fine-tuning

In a second step, we chose to use a pre-trained model of the BERT type (Bidirectional Encoder Representations from Transformers) (Jacob Devlin, 2019). BERT is built on the Transformer architecture and is pre-trained for language modelling.

Excellent results for many natural language processing tasks can be achieved by continuing to train this type of model. This can be accomplished by simply replacing the final layers of the model with layers appropriate to the task at hand and continuing to train the model weights from those pre-trained for language modelling.

In particular, we used DistilBERT (Victor Sanh, 2019), which is a distilled version of BERT that offers good performance for a significantly reduced number of parameters. This allowed us to reduce the training cost, which remains important for a model of this magnitude even for fine-tuning.

We thus replace the output language modelling module by a labeling head, i.e. a feedforward with two outputs. One trick that worked well in our case was to decrease the learning rate in the early layers of the model, which would extract more basic information that we would like to avoid re-entraining.

2.3 Detect answerability

After performing the BERT fine-tuning model on SQuAD 1, we wanted to apply it on SQuAD 2 which contains questions that have no answer in the associated context. To take into account this new constraint, we have indicated the first token of the sequence as a special token that must be pointed out by the model when the context does not contain an answer to a question. Since around 30% of the training data are unanswerable question, the model tended to constantly predict this special token. To counter this and be able to correctly answer answerable questions, we weight the loss differently to penalize more errors on answerable questions than errors on unanswerable questions.

3 Results

3.1 Qualitative results

The outputs of the model can first be evaluated qualitatively. We have reported several outputs of the fine-tuned BERT below.

When did Beyonce start becoming popular? ***Answer****: in the late 1990s.* ***Predicted****: late 1990s*

In what country is Normandy located? ***Answer****: France* ***Predicted****: France*

3.2 Quantitative results

To quantitatively evaluate the performance of a Question-Answering model, we computed Exact Matching and F1-score on validation data. Exact Matching is the proportion of answers extracted by the model that correspond exactly to the expected answer. F1-score is the harmonic mean between precision (the number of words in common between the expected answer and the prediction divided by the total number of words in the prediction) and recall (the number of words in common divided by the total number of words in the expected answer).

Data	Epochs	F1	EM	Time
Glove + Transformer				
Squad 1	5	0.16	0.11	22min
Fine-tuned DistilBERT				
Squad 1	1	0.77	0.67	1h18
Squad 2	1	0.44	0.39	1h33

4 Discussion

4.1 Fine-tuning or training from scratch

Clearly, BERT-based model's fine-tuning was much more effective than training from scratch. The representations obtained for language modelling are a great starting point for the question answering task. Although fine-tuning is far from instantaneous due to the huge number of model parameters, it only takes one epoch on the dataset to get very good results.

4.2 QA task with answerability

The addition of unanswerable questions was a challenge. Other implementations could be considered to adapt the model: a separate feedforward network could be trained to predict whether the question is answerable. Many state-of-the-art systems also use a verifier, such as a second transformer that will assign a probability to the context/question/answer triplet found.

References

Konstantin Lopyrev Percy Liang Pranav Rajpurkar, Jian Zhang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv:1606.05250*.

Percy Liang Pranav Rajpurkar, Robin Jia. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv:1806.03822*.

Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

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