Question Answering on SQuAD Dataset

Authors:
Angely Jazmín Oyola Suárez
Claudio Bonetta

Introduction **2**

Execution **4**

Loading the dataset 4

Reorganizing the dataset 4

Splitting the dataset 5

Training pre-processing 5

Handling Long Sequences 6

Training 7

Evaluation pre-processing 7

Prediction 7

Evaluation post-processing 8

Error Analysis **9**

Limitations and Improvements **13**

References **14**

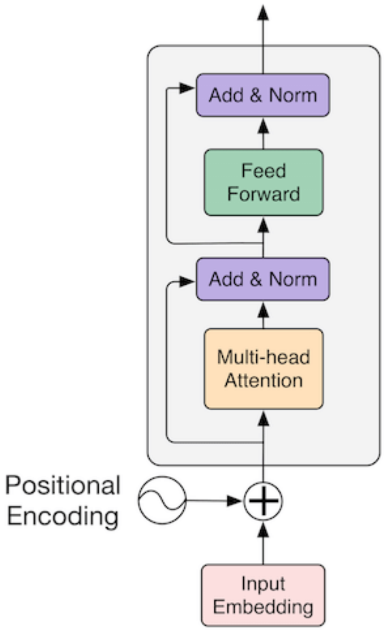
Introduction

The first step that has been done in order to carry out the project is the task assessment and the choice of the most appropriate methodology.

As such, having to deal with a question answering kind of task, it has been chosen to use a transformed-based model, since they constitute the current state of the art for the said task.

Since we are dealing with a non-generative question answering task, an encoder transformer architecture has been deemed to be the most appropriate.

The generic representation of an encoder transformer architecture can be found in the picture on the right (presented in the paper “attention is all you need”[1]).



Since these architectures are very complex both to build and to train, in order to define and train the model we have relied on a library called HuggingFace[2][3].

This library facilitates operating with transformers-like architectures by allowing to use pretrained models and easily fine-tune them on the target task.

Since, as previously mentioned, the training of these architectures is quite complex, we have decided to use a pre-trained model. Specifically, the model that has been chosen is DistilBERT[4], which is a smaller version of the well known BERT[5] and on which has been applied a knowledge distillation technique during the pre-training phase, using BERT as the teacher network. This fits our case, since the model has been trained with limited computing capabilities (using Google Colab).

Since DistilBERT has been pre-trained on a task different than question answering (masked language modeling and next sentence prediction), a fine-tuning step was required.

In particular, the model has been fine-tuned on the dataset (splitted as per requirements) given in the project’s material for three epochs, each of which took about one hour.

The main challenges posed by this task are:

- The model's input is composed by the concatenation of question and context as such, being aware of their position inside of the tokenized input sequence while working on it is not trivial. This is especially evident not only during the preprocessing phase but also in the postprocessing, since the truncation on the context requires evaluating for each question-context pair all the possible candidate solutions generated by each question-truncated context pairs (given in turn by the original question-context pair).
- The contexts are long, as such emerges the necessity to truncate them (transformers-based models have a maximum input sequence length that they can manage). A simple truncation is not possible since it would possibly exclude part of the answer. This problem has been approached by "sliding" the context over the model allowing some overlap, generating from a single pair question-context multiple pairs of question-truncated context.
- The training time is long: one epoch takes about one hour (with the GPU provided by google colab).
- Model's predictions need to be checked for consistency (explained better in the post-processing chapter).

Execution

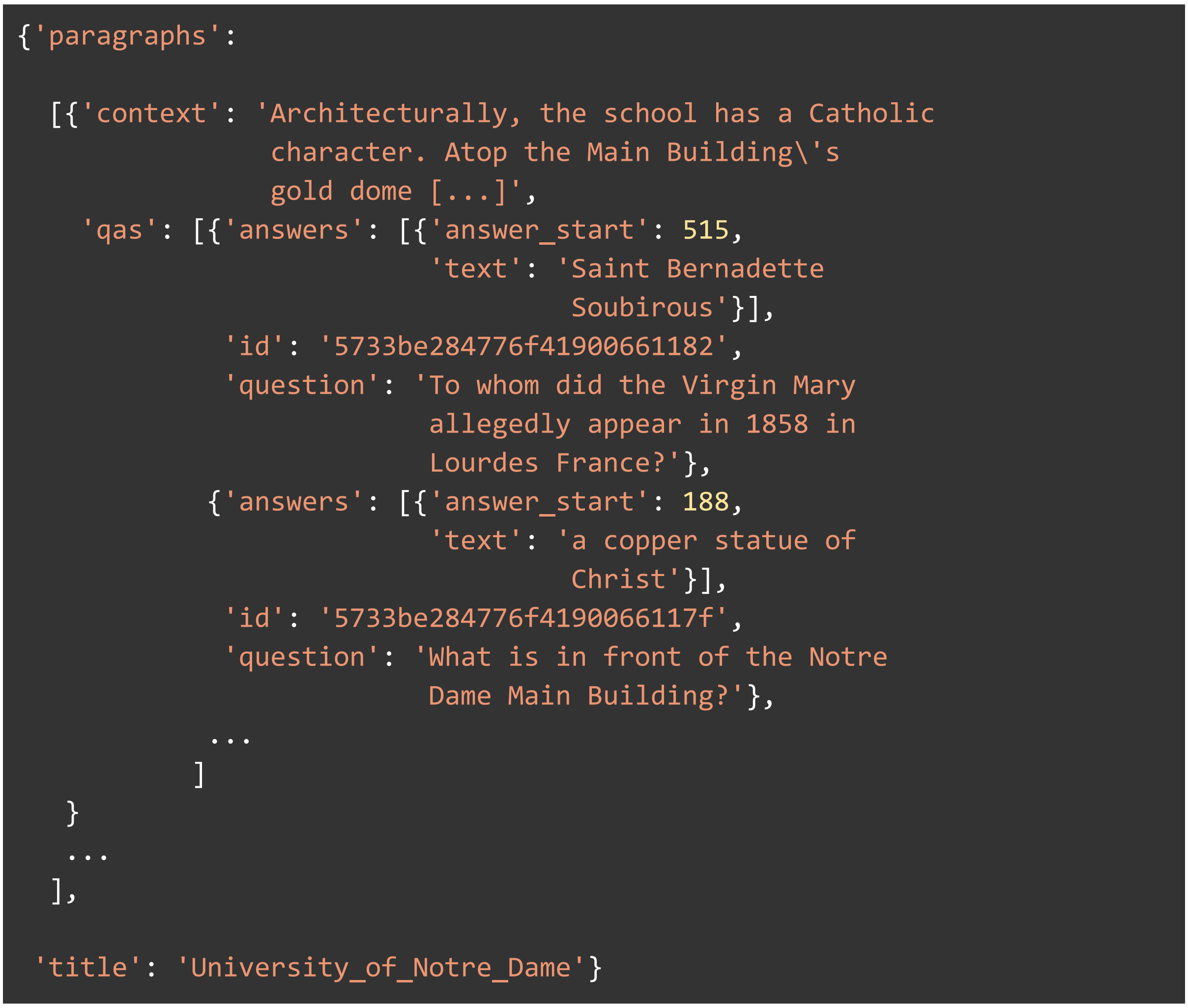
In this section will be listed and explained the approaches used to carry out the project.

Loading the dataset

The first steps were reorganizing the dataset in a way that would have been easily usable by the model and then splitting it conformingly with the project's requirement.

Reorganizing the dataset

The dataset's rows that has been given as project material looked like the following:

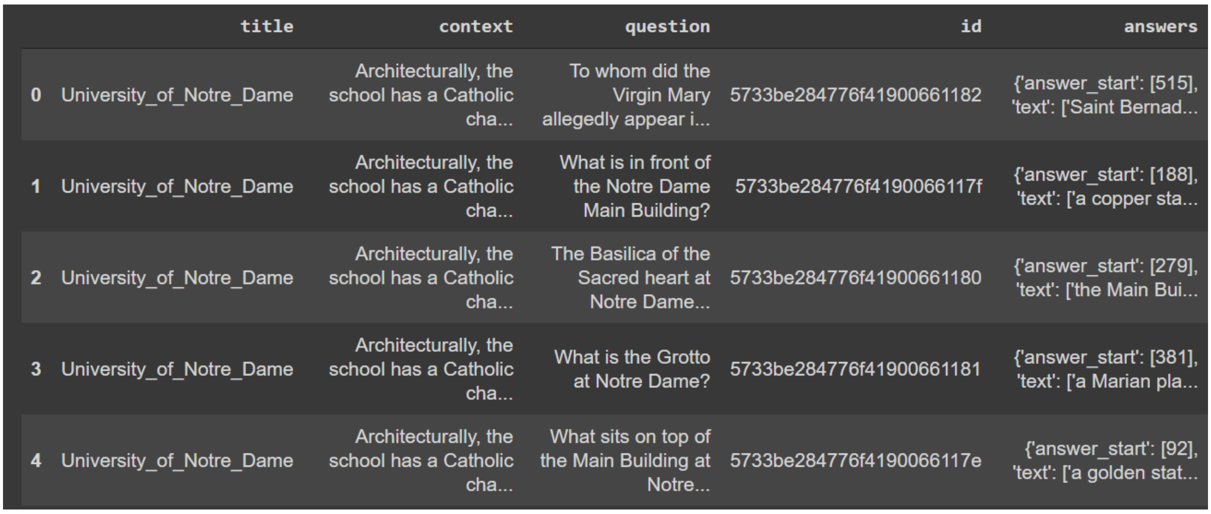


```
{'paragraphs':

  [{'context': 'Architecturally, the school has a Catholic
               character. Atop the Main Building\'s
               gold dome [...]',
    'qas': [{'answers': [{'answer_start': 515,
                          'text': 'Saint Bernadette
                                   Soubirous'}],
             'id': '5733be284776f41900661182',
             'question': 'To whom did the Virgin Mary
                          allegedly appear in 1858 in
                          Lourdes France?'},
            {'answers': [{'answer_start': 188,
                          'text': 'a copper statue of
                                   Christ'}],
             'id': '5733be284776f4190066117f',
             'question': 'What is in front of the Notre
                          Dame Main Building?'},
            ...
           ]
   }
   ...
  ],

 'title': 'University_of_Notre_Dame'}
```

This kind of structure, though, is not properly fitted to the task that we have at hand (or to phrase it better, it is not fitted to the kind of model's architecture we are using).
As such what has been done is to frame the dataset in such a way that each title has associated a single context and a single question/answer as well.
This implies a replication of the titles (since one title has more contexts) and context (since one context can hold more than one question/answer).
The final result is as follows:



	title	context	question	id	answers
0	University_of_Notre_Dame	Architecturally, the school has a Catholic cha...	To whom did the Virgin Mary allegedly appear i...	5733be284776f41900661182	{'answer_start': [515], 'text': ['Saint Bernad...
1	University_of_Notre_Dame	Architecturally, the school has a Catholic cha...	What is in front of the Notre Dame Main Building?	5733be284776f4190066117f	{'answer_start': [188], 'text': ['a copper sta...
2	University_of_Notre_Dame	Architecturally, the school has a Catholic cha...	The Basilica of the Sacred heart at Notre Dame...	5733be284776f41900661180	{'answer_start': [279], 'text': ['the Main Bui...
3	University_of_Notre_Dame	Architecturally, the school has a Catholic cha...	What is the Grotto at Notre Dame?	5733be284776f41900661181	{'answer_start': [381], 'text': ['a Marian pla...
4	University_of_Notre_Dame	Architecturally, the school has a Catholic cha...	What sits on top of the Main Building at Notre...	5733be284776f4190066117e	{'answer_start': [92], 'text': ['a golden stat...

Splitting the dataset

After the dataset's reorganization it has been splitted in train and validation sets based on row's titles with a proportion 9-1.

Training pre-processing

The preprocessing phase that comes before training has the aim of preparing the input and the labels that will be fed to the model during the training procedure.
The input consists of the question and the context, while the output will be the answer position inside the context (to be more precise it's the likelihoods for each index of the input sequence to be the answer's starting and ending index).

The first step would be concatenating and tokenizing the input (composed by the question and the context).

This implies that when working on the input sequence distinguishing between question and context is more difficult. Still, the tokenizer provides a tagging sequence that discerns, for each token, whether they belong to the question or the context (or if it is a special token).

The tokenization process has to faithfully replicate the one done during the pre-training phase (this of course includes the algorithm and the used dictionary) as such, defining which model is going to be used, becomes necessary since the preprocessing step. For this project we used DistilBERT for the reasons mentioned in the introduction.

Then, the labels also need to be computed: the dataset gives the starting character and the answer's text (through which we can find the ending character) but the model's need the answer's starting and ending token's index inside of the tokenized input sequence.

Handling Long Sequences

There's a problem concerning the kind of data we had to use and the task they had to be used for: the context is, oftentimes, too long for the model to handle.

In fact transformers-based models have a maximum length that they are able to handle. The solutions when handling long sequences are usually two:

- Choose a model that is capable of handling long sequences (such as Longformer[6]).
- Truncate the input sequence.

We have opted for the latter.

Vanilla truncation, while it may still be an effective approach for a number of tasks, in this particular case it does not constitute a valid option since it may throw away part of or even the whole answer.

So, what has been done to solve this issue is "sliding" the context over the model. As a consequence, instead of having a pair question-answer, after this procedure we would have multiple pairs of question-truncated context.

This brings some difficulties when detecting the answer starting and ending token inside the tokenized input since the start of the context inside the tokenized input sequence does not necessarily correspond to the effective starting point of the non-truncated context. This is however solved by relying on the information provided by the tokenizer which saves for each input token (of every question-truncated context pairs) their starting and ending character’s index inside their respective original string (of either the question or the context).

Training

Since DistilBERT has originally been pre-trained to be used on masked language modeling and next sentence prediction, it is necessary to attach to the backbone of the model a new head fit for a question-answering task and then fine-tune the model.
The model has been fine-tuned for three epochs, each of which lasted for about one hour.

Evaluation pre-processing

The preprocessing procedure done before the evaluation is slightly different with respect to the one done before the training.
The main difference is the aim of this procedure: while the one done before the training has as aim the computation of the labels, this one simply prepares some data that will be useful for the post-processing phase. Still, since after this procedure comes the prediction, the questions and contexts need to be tokenized the same way as it has been done during the train’s pre-processing.

Prediction

The model’s prediction function takes as arguments the preprocessed evaluation data and gives as output for each data row the likelihood of each input sequence’s token to be the answer’s starting and ending token.
Since the output of the model consists only of likelihoods, a post-processing step in which the textual answer is retrieved is required.

Evaluation post-processing

The aim of the post-processing function is to convert the likelihoods given as output by the prediction into a textual answer.

This process is not as immediate as taking the maximum likelihood's index as the answer's starting and ending token's index and this is mainly for two reasons:

- The starting index may result in being after the ending index.
- Since the input is composed of both the question and the context, the starting and/or the ending index may fall inside the question.

So what has been done is choosing the pair of indexes whose score is maximum (the score is given by summation of the starting and ending indices' likelihood) and that respect the consistency constraints mentioned above.

Since creating and comparing the score between all the couples of starting and ending index have a quadratic complexity (with respect to the length of the tokenized input sequence) only a small portion of the likelihoods has been used to compute the score (around 15% of the indexes which associated likelihoods is the maximum among the starting/ending answer's token's index likelihood).

Additionally, a pair question-context may generate multiple pairs question-truncated context this implies that more than one candidate may be generated for a single input. In order to select the best candidate the one which has the highest score has to be taken.

Once the answer's starting and ending token's index have been selected, what's left is to retrieve the corresponding portion of context's text.

Outcome Analysis

Performances

The metrics used to evaluate the model are exact match and f1-score as per suggestion of the original SQuAD paper[7]. The results obtained over the test set are 69.43 for the exact match and 80.70 for the f1-score. For reference one of the best performing models on this dataset is LUKE which scores 89.8 over the exact match and 95.0 over the f1-score[8].

Errors

A sample of the mistakenly answered questions has been analyzed. In this case with “mistakenly answered” we will be referring to those normalized predicted answers that do not exactly match the normalized ground truth. Normalization consists of lowercansing**,** removing punctuation, removing articles and stripping extra white spaces.

What has been observed is that many of the mistakenly predicted answers, even though they do not exactly match the ground truth, they still share a high degree of similarity and can be said that they effectively answer the given question.
Here’s a case in which the answer is a superset of the ground truth (but still correct):

> **Question**: What did Gao Qiang tell reporters in Beijing?
> **Context**:
> [...] Vice Minister of Health Gao Qiang told reporters in Beijing that the "public health care system in China is insufficient." [...]
> **Ground truth**: public health care system in China is insufficient
> **Predicted**: that the "public health care system in China is insufficient."

Here instead the answer is a subset of the ground truth (but still correct):

> **Question**: How many highways leading into Wenchuan were damaged?
> **Context**:
> All of the highways into Wenchuan, and others throughout the province, were damaged, resulting in delayed arrival of the rescue troops. [...]
> **Ground truth**: All of the highways
> **Predicted**: All

Sometimes the predicted answer can be considered correct even though it's completely different from the ground truth:

> **Question**: The previous record beaten by Park Avenue was for what real estate?
> **Context**:
> [...] 450 Park Avenue was sold on July 2, 2007 for US\$510 million, about \$1,589 per square foot (\$17,104/m^2), breaking the barely month-old record for an American office building of \$1,476 per square foot (\$15,887/m^2) set in the June 2007 sale of 660 Madison Avenue. [...]
> **Ground truth**: 660 Madison Avenue
> **Predicted**: American office building

In this case the ground truth specifies the real estate's address, while the prediction indicates the type of building. Since the question asks only "what real estate", the predicted answer can be considered to be valid as well.

While in the case of the previous examples the answer, even though different, still succeed in properly answering the question, there are some cases where, even though the answer is included, presents some neighbouring context unrelated to the question:

> **Question**: What publications were shut down 1972?
> **Context**:
> [...] Ruling by decree, the RCC maintained the monarchy's ban on political parties, in May 1970 banned trade unions, and in 1972 outlawed workers' strikes and suspended newspapers. [...]
> **Ground truth**: newspapers
> **Predicted**: workers' strikes and suspended newspapers

In this last example what probably happened is that the model fails to understand that "newspapers" is an instance of "publications" and focuses on "shut down" and "1972" taking what comes after "1972 outlawed". This should show the capability of the model of associating "outlawed" with "shut down": if only "1972" were to be captured then maybe the word "outlawed" would have been included in the answer as well.

Another interesting example of this category is:

> **Question**: In what year was the article describing the type of arsenic found in Napoleon's hair published?
> **Context**:
> [...] According to a 2007 article, the type of arsenic found in Napoleon's hair shafts was mineral, the most toxic, and according to toxicologist Patrick Kintz, this supported the conclusion that he was murdered.
> **Ground truth**: 2007
> **Predicted**: 2007 article, the type of arsenic found in Napoleon's hair shafts was mineral

This happened probably because the model paid attention to the question's tokens "what year" and "type of arsenic found" failing to understand that what was being asked was only the year. The fact that the two information are close in the context is probably another factor that contributed to this outcome.

Another similar example:

> **Question**: What is The Triple Entente?
> **Context**:
> [...] The Triple Entente was the name given to the loose alignment between the United Kingdom, France, and Russia after the signing of the Anglo-Russian Entente in 1907. [...]
> **Ground truth**: the name given to the loose alignment between the United Kingdom, France, and Russia after the signing of the Anglo-Russian Entente in 1907.
> **Predicted**: the loose alignment between the United Kingdom, France, and Russia

While in this case the prediction does not completely fail to capture the ground truth, the quality of the answer is debatable: when explaining an historical event an human speaker would also expect an historical collocation of such an event. What probably happens is that in this case the model fails to understand this subtle detail and only focuses on "what" (in the question), which doesn't generally imply a chronological knowledge (which is generally expressed using "when").

Sometimes the model just gives errors.
One of the reasons for such behaviour may be that the question is posed in a difficult way.

Question: What did the court determine was the fate for Maclean after shooting the Queen?

Context:

[...] Victoria was outraged when he was found not guilty by reason of insanity, but was so pleased by the many expressions of loyalty after the attack that she said it was "worth being shot at—to see how much one is loved".

Ground truth: not guilty by reason of insanity

Predicted: to see how much one is loved

In this case, in order to properly answer this question, the model should understand that “fate” is somehow connected to “guilty”.

Limitations and Improvements

For starters a consideration about the evaluation metrics can be done: the exact match, even though a normalization procedure is applied, risks of not counting answers that are still correct. Another relevant metric that can be introduced is IOU which would be able to capture in greater detail the complexity of the prediction with respect to the ground truth: for example, the question “In minutes, how long does it take for the average New Yorker to get to work?” having as ground truth “38.4” and a prediction “38.4 minutes a day” would still be somewhat counted.

Generally speaking the reported limitation could be said to include all of those predictions that are subset/superset of the ground truths and that still faithfully answer the question.

Another way to address this problem would be, for each question/context, having more possible candidate answers. This limitation of the dataset has also been reported in SQuAD’s original report[7].

Another limitation is that DistilBERT has a quadratic time complexity typical of many transformers-based architectures and the fact that it is only able to handle a maximum of 512 tokens.

Both those problems may be addressed using Longformer which is a transformers-based architecture model that works in linear time complexity and that is able to handle longer input sequences with respect to DistilBERT.

The used approach (based on truncation) has been preferred with respect to using a model that handles longer sequences simply because it is more generalizable (even a model that handles longer sequences has limitations, even though higher).

References

[1]: VASWANI, Ashish, et al. Attention is all you need. En Advances in neural information processing systems. 2017. p. 5998-6008.

[2]: WOLF, Thomas, et al. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771, 2019.

[3]: https://huggingface.co/

[4]: SANH, Victor, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[5]: DEVLIN, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[6]: BELTAGY, Iz; PETERS, Matthew E.; COHAN, Arman. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150, 2020.

[7]: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761899.pdf

[8]: Yamada, Ikuya, et al. "Luke: deep contextualized entity representations with entity-aware self-attention." arXiv preprint arXiv:2010.01057 (2020).