Question Answering on SQuAD dataset

Marco Aspromonte
marco.aspromonte@studio.unibo.it

Valentina Boriano
valentina.boriano@studio.unibo.it

Enrico Morselli
enrico.morselli@studio.unibo.it

Francesco Romito
francesco.romito2@studio.unibo.it

January 2022

Abstract

The aim of this project is to develop a question answering model (QnA) on *SQuAD* dataset.

We achieve it using transformers, whose high performances in the field of natural language processing are well-known. In particular, we focused on **distil-BERT**, **ALBERT** and **distil-RoBERTa** in order to find the relationship between the transformers'best results and its complexity. We studied the models'results compared to the dataset, in fact we have noticed that the length of the answer and of the question and the type of question affect a lot the accuracy of the transformer. Using the F1-score as our metric, we find out that the *ALBERT* is the most accurate model. Distil-BERT and distil-RoBERTa have great outcome but less precise than *ALBERT*. Finally, we decided to test our models on a split of SQuAD Dataset and on ***DuoRC***[3] dataset (that was the most compatible to SQuAD).

The code can be found on our GitHub repository of the project [7].

Contents

1 Introduction

Question Answering is the task of extracting a span of text from a given context paragraph as the answer to a specified question. This assignment has seen considerable progress in recent years with applications in search engines.

First of all, we analyzed the *SQuAD* dataset, in order to have a more complete information of it. Then we divided it in training set and validation set (respectively 75% and 25%). To solve the task, we utilized 3 types of transformers, that are: DistilBERT, DistilRoBERTa and ALBERT. To compare the quality of the models we utilized the F1 metric and the exacting match measure. In the end, we tested our models with the validation split of *SQuAD* and for the sake of completeness of the research, we tested them also with another dataset **DuoRC**.

In this project, we studied the difference between various pretrained transformer-based language models to analyze how well they are able to generalize on *SQuAD* dataset. In our report we will discuss our approach with the transformers that we have utilized, how is composed the dataset, the obtained results and how the dataset and the model complexity affects its accuracy.

2 Transformers

The Transformer in NLP is a new architecture that aims to solve sequence-to-sequence tasks. Transformers are manly divided in two part: Encoder and Decoder.

Each encoder layer contains two sub-layers:

- the first sub-layer is Multi-Head Attention.
- the second sub-layer is Feed Forward Neural Network

The Decoder is composed by 3 sub-layers:

- the first sub-layer is Multi-Head Attention.
- the second sub-layer is Feed Forward Neural Network
- the third sub-layer is another Multi-Head Attention layer.

To implement transformers we used Tensorflow.

To solve the Question and answering task we decided to test different transformers such as: DistilBERT, ALBERT and DistilRoBERTa. In this section we will explain the main differences between them and in the section 4 we will show the outcome for each.

2.1 DistilBERT

DistilBERT, a distilled version of BERT, is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased (so, 66M parameters), runs 60% faster while preserving over 95% of BERT's performances.

2.2 ALBERT

ALBERT simply utilizes the BERT architecture. This architecture, which itself is the encoder segment from the original Transformer (with only a few minor tweaks). It is changed in three key ways, which bring about a significant reduction in parameters (11M parameters):

- **Key difference 1**: embeddings are factorized, decomposing the parameters of embedding into two smaller matrices in addition to adaptations to embedding size and hidden state size.
- **Key difference 2**: ALBERT applies cross-layer parameter sharing. In other words, parameters between certain subsegments from the (stacked) encoder segments are shared, e.g. the parameters of the Multi-head Self-Attention Segment and the Feedforward Segment. This is counter to BERT, which allows these segments to have their own parameters.

2.3 Distil RoBERTa

This model is a distilled version of the RoBERTa-base model. It follows the same training procedure as DistilBERT. The code for the distillation process can be found here. This model is case-sensitive: it makes a difference between english and English.

The model has 6 layers, 768 dimension and 12 heads, totalizing 82M parameters (compared to 125M parameters for RoBERTa-base). On average DistilRoBERTa is twice as fast as Roberta-base.

We encourage to check RoBERTa-base model to know more about usage, limitations and potential biases.

2.4 Optimizer

We created a custom optimizer with the following values:

- learning rate: 2e-5 (we found other admissible by suggestion are 5e-5 and 3e-5 [**?**]
- batch_size = 2
- num_train_epochs = 2 for DistilRoBERTa and 3 for DistilBERT and ALBERT
- weight_decay = 0.01
- max_length = 384 The maximum length of a feature (question and context), it's a standard value
- doc_stride = 128 The authorized overlap between two part of the context when splitting it is needed.

3 Data

3.1 SQuAD Dataset

We utilized the Stanford Question Answering Dataset*SQuAD* that is a reading comprehension dataset made up of questions posed by crowd workers on a collection of Wikipedia articles, with the response to each question being a text segment, or span, from the relevant reading passage.

The reading sections in SQuAD are taken from high-quality Wikipedia pages, and they cover a wide range of topics from music celebrities to abstract notions. A paragraph from an article is called a passage, and it can be any length. Reading comprehension questions are included with each passage in SQuAD. These questions are based on the passage's content and can be answered by reading it again. Finally, we have one or more answers to each question.

One of SQuAD's distinguishing features is that the answers to all the questions are text portions, or spans, in the chapter. These can be a single word or a group of words.

In our project, we divided the dataset in training and validation set (respectively 75% and 25% of SQuAD). Our dataset is composed by 5 features:

- Title
- Id
- Context
- Question
- Answer

Here, we show an example of it:


```
{'answers': {'answer_start': [515], 'text': ['Saint Bernadette Soubirous']},
 'context': 'Architecturally, the school has a Catholic character. Atop the Main Building\'s gold dome is a golden statue of the Virgin Mary. Immediately in front
 'id': '5733be284776f41900661182',
 'question': 'To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?',
 'title': 'University_of_Notre_Dame'}
```


Figure 1: Example of the dataset

Then we studied how the dataset is composed in order to have a better error analysis on the results.

First of all, we show the distributions of the lengths in words of answer, question and text on train and test set. Then we checked the distribution of the types of questions. Below are the graphics:

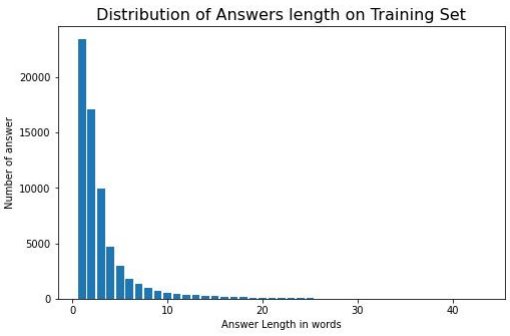

Figure 2: Distribution of the answers' length(in words) on training set

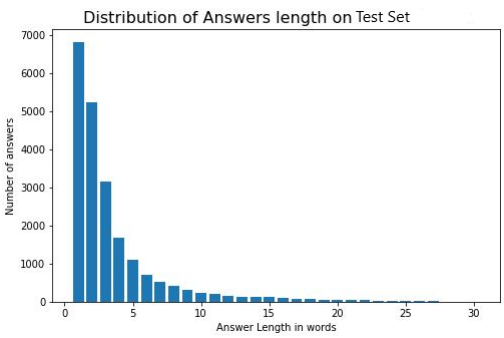

Figure 3: Distribution of the answers' length(in words) on test set

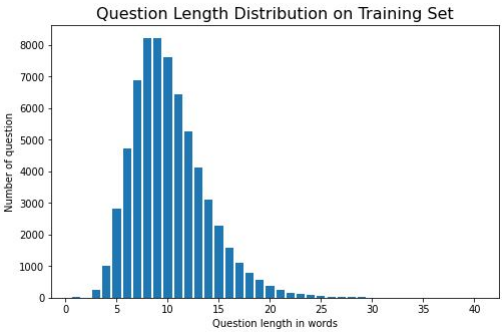

Figure 4: Distribution of the questions' length(in words) on training set

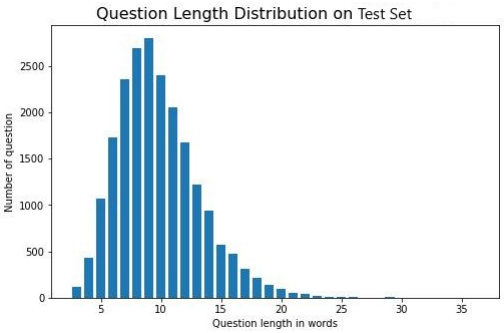

Figure 5: Distribution of the questions' length(in words) on test set

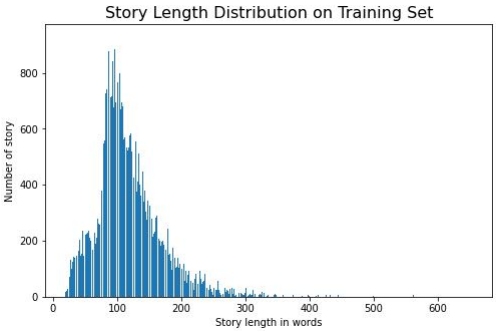

Figure 6: Distribution of the context's length(in words) on training set

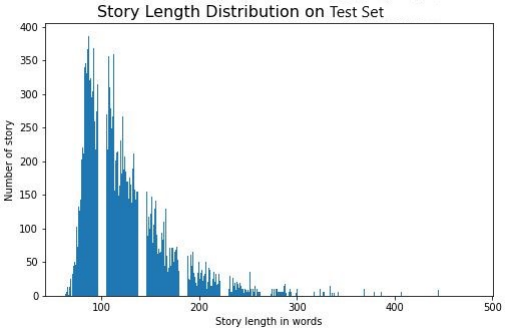

Figure 7: Distribution of the context's length(in words) on test set

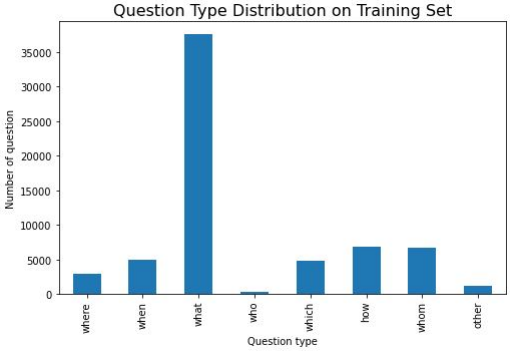

Figure 8: Distribution of questions' type on training set

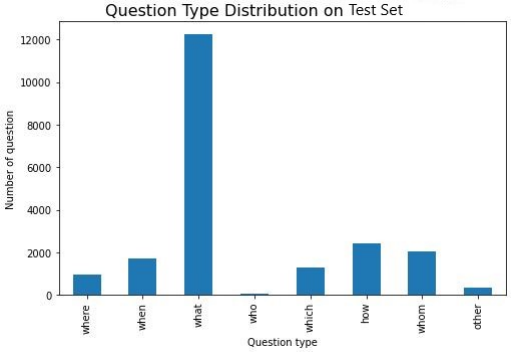

Figure 9: Distribution of the questions' type on test set

We can see that *SQuAD* dataset on training and test set is well distributed. In fact, it contains, in both, homogenous data for answers, question and context. Due to that the predictions will not be affected by the diversity of training and test set.

3.2 DuoRC Dataset

DuoRC contains 186,089 unique question-answer pairs created from a collection of 7680 pairs of movie plots where each pair in the collection reflects two versions of the same movie. This dataset is distributed as follows:

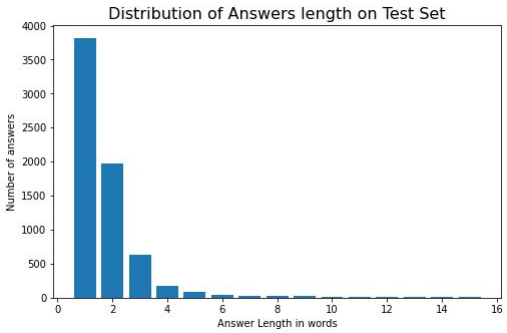

Figure 10: Distribution of answers' length (in words) on DuoRC

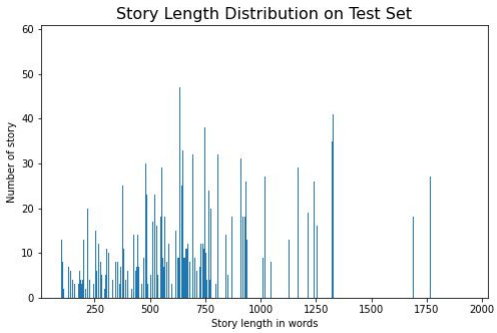

Figure 11: Distribution of the stories' length (in words) on DuoRC

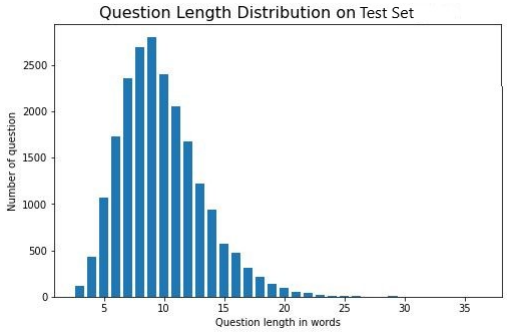

Figure 12: Distribution of question' length (in words) on DuoRC

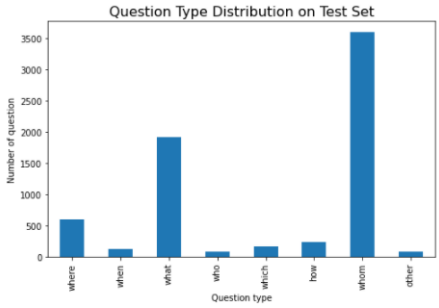

Figure 13: Distribution of the questions' type on DuoRC

In the figure is shown that, the distribution of the answers' length and the questions' length is very similar to the distribution of *SQuAD* dataset. The most important differences are on the distribution of the length of the context and the type of the question that fortunately are less affecting than the others.

We decided to utilize this dataset as an additional test set for the models. The results were surprisingly accurate, in fact the F1 score is very high, reaching a maximum of 80% using the ALBERT model.

3.3 Tokenization

Then we proceed with the tokenization, because in order to be passed as an input to transformers, text must be tokenized, where each word is converted into integer. Tokenization is an important step because of a good choice for the tokenization could influence the quality of results. In particular, we have used the function tokenize_dataset, in which the following steps are deployed:

- Tokenize the input sentence
- Add the CLS (special classification token) and the separator tokens
- Pad or truncate the sentence to the maximum length allowed, in this case 384
- Manage the overlapping with the doc_stride set to 128

After, the tokenized datasets (train, validation and test) are converted to a TensorFlow format, creating the attention masks which helps to differentiate real tokens from padding tokens.

4 Results

In this section, we show the outcome for each transformer on the two different test set. We show the f1 score, the exact match that is the percentage of the correct predictions and the total matches. Following, we show the graphics of some model, that compare the F1 score and the property of the text previously described. In the end, we show some wrong prediction.

4.1 DistilBERT

DistilBERT has very impressive results, in fact it reaches very high F1 score on both test set. Below we show the outcomes. On both test sets it reaches the same f1 score. This is the fastest model in the training process, needing only 1h:40 per epoch on a GTX1650 Ti, which is in line with his medium-low number of parameters

4.1.1 SQuAD Test Set

```
exact: 59.967381174277726
f1: 75.43111278382322
total: 21460
```

4.1.2 DuoRC Test Set

```
exact Match: 67.34634430527235
f1: 75.18420794877258
total_matches: 6866
```

Now we show the F1 score average compared to the length of answer, question and context:

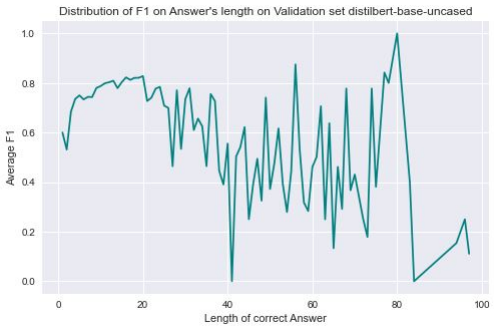

Figure 14: Average of F1 on answer's length

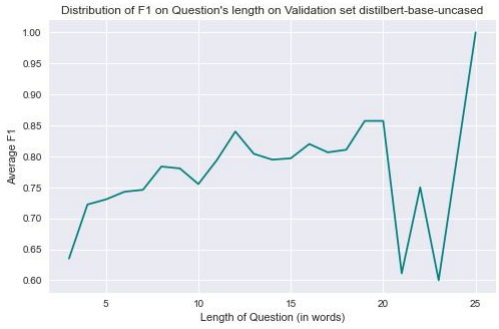

Figure 15: Average of F1 on question's length

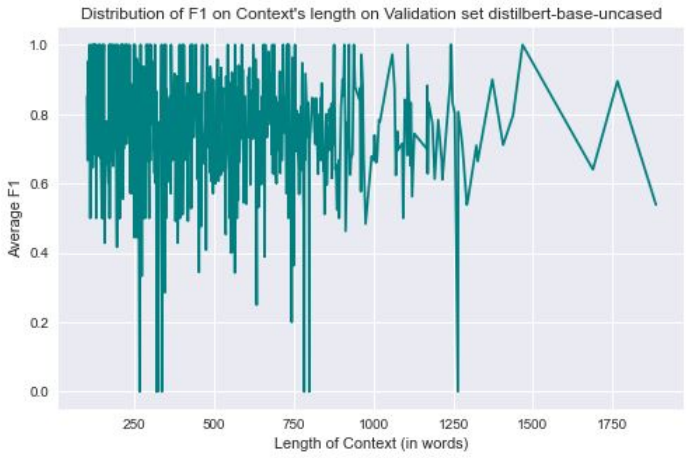

Figure 16: Average of F1 on context's length

In the figure above, we can see that on DuoRC test set, the f1 score compared with the property of text does not outline a specific influence of the dataset. The only important information is about the figure 14 that highlights how the answer's length affects the accuracy.

4.2 ALBERT

This model is the most accurate one, in fact, it reaches formidable results, gaining a f1 score almost equal to 80 in both tests set. Surprisingly ALBERT is the slower one in the training phase, despite his reduced number of parameters with 3h:15 per epoch on a GTX1650 Ti.

4.2.1 SQuAD Test Set

Below are showed the f1 score metric:

```
exact: 64.24044734389562
f1: 79.99211122229185
total: 21460
```

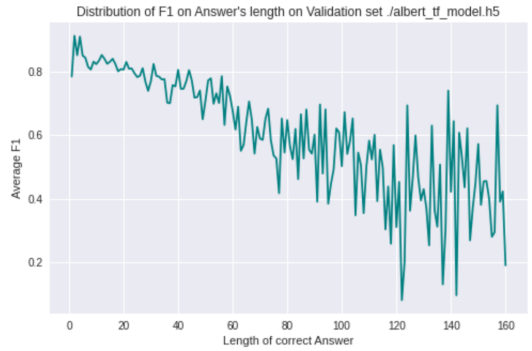

Figure 17: Average of F1 on answer's length

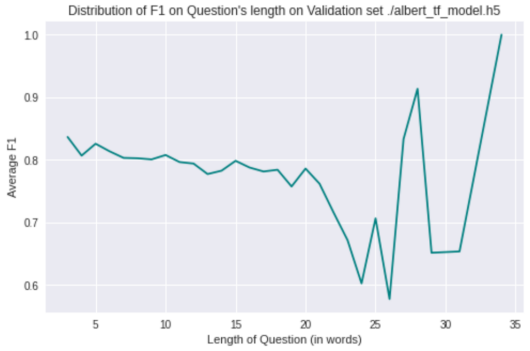

Figure 18: Average of F1 on question's length

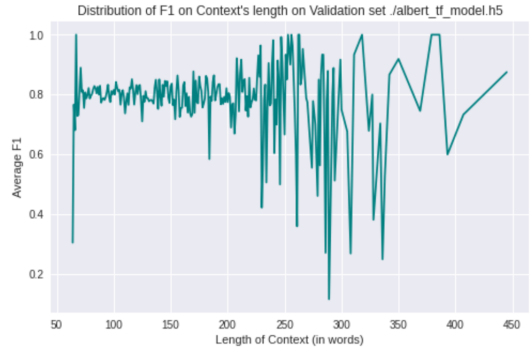

Figure 19: Average of F1 on context's length

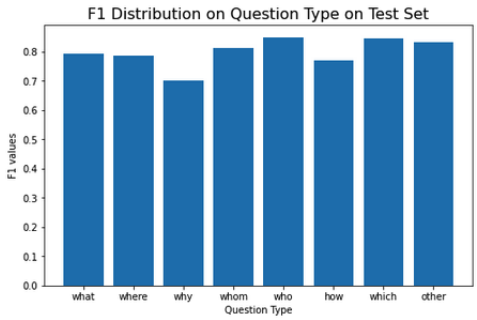

Figure 20: Average of F1 on question's type

4.2.2 DuoRC Test Set

```
exact: 72.32741042819691
f1: 80.35232092902714
total: 6866
```

4.3 Distil RoBERTa

DistilRoBERTa produced very great results, almost as ALBERT, in fact it reaches a f1 score near to 80 on SQuAD test set. But the DuoRC test set, the f1 score is near to 75. Despite this is the largest model, it is not the slowest in the training process, needing just 2h:05 per epoch on a GTX1650 Ti.

4.3.1 SQuAD Test Set

```
exact : 63.62068965517241
f1 : 78.7033207313422
total : 21460
```

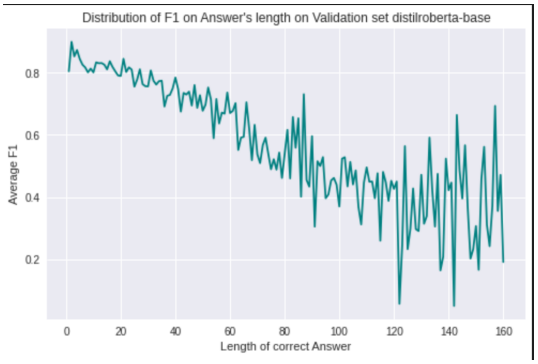

Figure 21: Average of F1 on answer's length

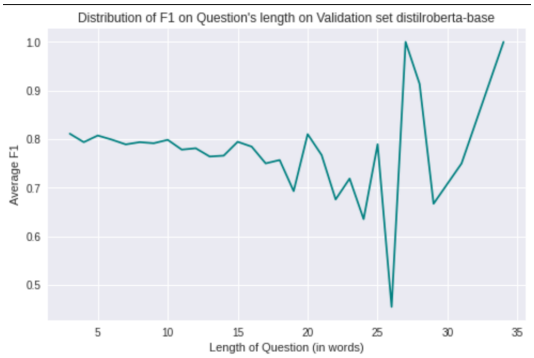

Figure 22: Average of F1 on question's length

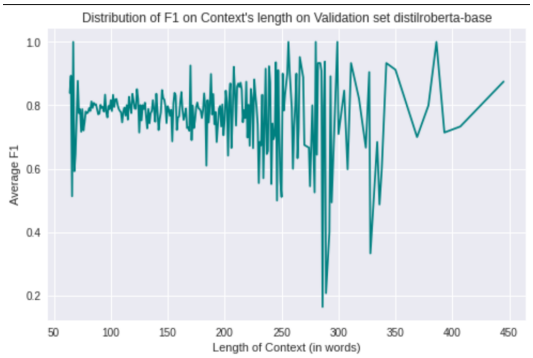

Figure 23: Average of F1 on context's length

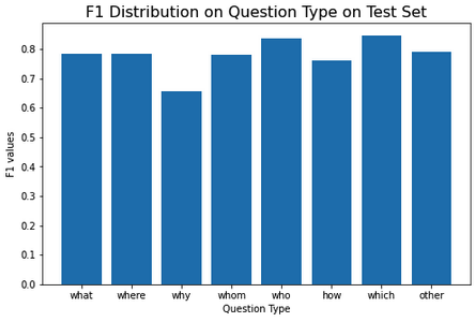

Figure 24: Average of F1 on question's type

4.3.2 DuoRC Test Set

```
exact: 65.97727934750947
f1: 74.47926991429914
total: 6866
```

4.4 Error Analysis

To check the power of our models, we plotted many graphics comparing the F1 score to different property of the test, such as: length of the answers, question and context and type of the question. We can see in figure 21 and 22 that the length of the answer and of question, affect a lot the accuracy of the models. Increasing the length of answers and questions, the f1 score quickly decrease. On the contrary, the length of the context does not influence the precision of the predictions. We argued that the descent of the f1 score could due to complexity of the text and not due to its length. Finally, as shown in the figure 24, we can see that, the type of question affect a lot the accuracy of the prediction, in fact the "why" question has a very low score comparing to the other types of questions.

Here we show an example of the predictions with the lowest F1 score in DistilBERT model with DuoRC as test set:

```
Context:  Based on a true story, Curtis Plummer (Ice Cube) is a down-on-his-luck former
          high school football star that turns his niece, Jasmine (Keke Palmer),
          into the quarterback of the local team, the Minden Browns.[...].She not only looks up
          to her uncle, but honors him by wearing his old jersey number.
Question:  What is the name of the local team?
Answer:  a man wearing a grinning translucent mask
Predicted answer:  Minden Browns

Context:  Based on a true story, Curtis Plummer (Ice Cube) is a down-on-his-luck former
          high school football star that turns his niece, Jasmine (Keke Palmer),
          into the quarterback of the local team, the Minden Browns.[...].She not only looks up
          to her uncle, but honors him by wearing his old jersey number.
Question:  What is the name of the local team?
Answer:  Santa Barbara, California, and Jake Adler
```

```
Predicted answer:  Minden Browns
```

These predictions show that in some cases the model pick a "random" piece of text trying to answer correctly.

5 Conclusion

BERT in general is a very powerful model for the QnA task, in fact from the results we can assert that the best accuracy is reached by ALBERT with a more or less 80% of f1-score, this is quite interesting because the trained h5 weights with ALBERT occupy approximately 50 MB, while with Roberta and Distilbert we have to consider approximately 300 MB. This is fancy because one of the most discussed objective is not only to find the best accuracy, but also to minimize the number of the parameters, taking into account for example that NVIDIA Megatron Transformers is about 1T of size to feed a model of 1 trillion parameters. Of course there are many differences between models like this or T5 and BERT, since they are able to solve many tasks with the pre-trained weights (such as translation) while BERT has only a language model.

A very interesting aspect is that BERT is quite versatile considering that we can use some of its transformers with a quite common GPU, even if some of the transformers are very expensive in terms of time and space, such as Bert Base or full-Roberta (16GB of size). So a good approach to achieve this task is to find a good ratio between time and space with respect to the available resources in order to choose the best architecture basing on them and requests of the problem.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019), *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, https://arxiv.org/abs/1810.04805.

[2] Kate Pearce, Tiffany Zhan, Aneesh Komanduri, Justin Zhan *A Comparative Study of Transformer-Based Language Models on Extractive Question Answering* (2021), Tensorflow, https://arxiv.org/pdf/2110.03142.pdf, consulted on January 2022.

[3] A Large-Scale Dataset for Paraphrased Reading Comprehension https://duorc.github.io/

[4] A distilled version of BERT: smaller, faster, cheaper and lighter https://arxiv.org/abs/1910.01108

[5] A Lite BERT for Self-supervised Learning of Language Representations https://arxiv.org/abs/1909.11942

[6] DistilRoBERTa base model https://arxiv.org/abs/1909.11942

[7] https://github.com/valentinaboriano/Natural-Language-Processing.git