EVALUATION OF DIFFERENT MODELS FOR QUESTION ANSWERING OF INDIVIDUAL QUESTIONS AND DIFFICULTIES

25.01.2022

Firat Saritas[1], **Mario Arteaga**[1], **Simon Staehli**[1]
[1]University of Applied Sciences in North-Western Switzerland (FHNW)

Abstract

In this report, three different models are compared for question answering. To train the models the data set Squad-V1 was used. For the comparison an individual collection of 15 texts and 3 questions per text was created. Eventually, to measure the performance of the models, the F1-score was used of tokenized output answers as the evaluation metric. In our analysis it was visible that the model output performances depended on the difficulties of the questions. Further we could state that the model performances were not stable to unanswerable questions, long and multi-span answers and the interpretation of the given context. In the evaluation one could see that the most heavy model Roberta performed best throughout all text and question difficulties.

Keywords: Natural Language Processing, Question Answering, Transformer

1 Introduction

This paper covers the evaluation of different question answering models, which were trained on equal data set Squad-V1. This data set contains text denoted as context and related questions to the texts, whereas the number of questions differ from context to context. This data set is common used as a data set for the training of question answering models [5]. Section 2 covers and explains what question answering exactly is and which network architectures are applied for this kind of task. Furthermore, the models architecture, which are used for the evaluation, are explained. Section 3 contains the methods, which are used to evaluate the chosen model. In this section the collected questions are explained as well as the selected metric for the evaluation. Section 4 focus on the investigations and observations of the model errors and comparisons between all models. Further, it covers correction, which could be performed to fix the error issues. In section 5 gives a brief summary and conclusion of this work.

2 Model

In this section the application of question answering and an overview and summary of the used architectures will be given.

2.1 Question Answering

As the name denotes question answering describes models and architectures, which aim to answer questions given certain contexts. These models identify word spans in the question context, which match best to the given question [6]. Many of the architectures used for question answering are based on the transformer architecture [11] i.e. BERT. Inputs are given as context and query (question). The inputs were mostly fed to a word embedding (i.e. GLOVE) and then forwarded to the actual model. The output of the model corresponds with a span of the context which exposes the right answer [6].

2.2 Model Selection

In the model selection process we focused on models which were available on the plattform Huggingface [12]. The main focus was on lightweight transformer models like Albert and Distilbert. We also included Roberta which is not a lightweight model for the comparisson [8, 4, 5]. We have successfully used the Distilbert model in previous tasks.

2.3 BERT

BERT is a model based on the Transformer or vanilla Transformer architecture, proposed by Vaswani et al. in 2017 [11]; and as such, there are other approaches that work with this architecture in performing NLP tasks for example, the GPT-based models. This architecture essentially provides better way to parallelism of the training process, thus it is used in most of the newer architectures [11]. Next to it is a figure of the transformer model architecture 1.

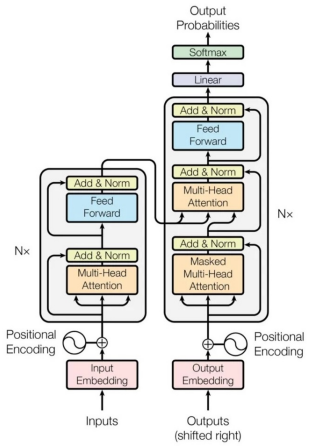

Figure 1: Proposed transformer architecture in the paper "Attention is all you Need"[11]

The transformer approach is composed of two elements, an encoder and a decoder like most of the sequence-to-sequence models. The encoder segment converts the original sequence into a hidden and intermediary representation, whereas the decoder segment converts this back into

a target sequence. The learning process of these models is unidirectional, from left to right [11].

The key blocks which were introduced at this time were the attention blocks, denoted as Multi-Head Attention in figure 1. The attention blocks allow the output components of a previous layer to attend to all positions from the output of the previous layer [11, p.5]. This is likely to the forwarded hidden states of sequential models for instance Simple RNN or LSTM, though each cell can attend to previous cells at all time, whereas the cells are limited in knowledge.

As the transformer architecture provides the basic building blocks for straight forward models (i.e. BERT or GPT), these are often used in most of NLP use-cases [12]. These models are trained on large corpus and can be fine-tuned on specific tasks like question answering, where they deliver good performances. The answer of the good results can be directly obtained by looking at the answer of Devlin et al.:

"We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training."[3]

"For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention layers of the Transformer [...] Such restrictions are sub-optimal for sentence-level tasks, and could be very harmful when applying fine-tuning based approaches to token-level tasks such as question answering, where it is crucial to incorporate context from both directions."[3]

The why is related to the context that is provided to a token during processing. During pre-training, unidirectionality in language models is not of much concern, given the training task performed by GPT during pre-training. But during fine-tuning, the problem becomes more clear. If we want to fine-tune to a specific task, not only previous tokens become important, but also future tokens with respect to some token will be.

That is why Devlin et al. (2018) argue why previous Transformers underperform, and why bidirectional language models perform better in a more natural language way [3].

BERT is a "language representation model" [3] and stands for "Bidirectional Encoder Representations from Transformers" and is currently the state-of-the-art for machine learning when it comes to NLP tasks. BERT was developed to pre-train deep bidirectional representations of unlabeled text by jointly joining the context in all layers. This allows the pre-trained BERT model to be refined with only one additional output layer. Starting with BERT, many other models have evolved. We perform our analyses with three of them: AlBert, DistilBert and RoBerta.

Albert stands for "A Lite BERT for Self-supervised Learning of Language Representations" [4] and was developed by Lan et al. [4]. This model offers two main techniques to reduce storage space and increase the speed of training. One of the techniques is to divide the embedding matrix into two small matrices and the other technique is to divide repeating layers into groups.

Distilbert stands for "a distilled version of BERT" [8]. So this model is smaller, faster,

cheaper and lighter. This model was designed by Sanh et al. [8]. The quantity of Distilbert parameters was reduced by 40% by using the principle of knowledge distillation. The Language Understanding could be maintained with 97%. In addition, the Transformer model was speeded up by 60% [8].

Roberta stands for "A Robustly Optimized BERT Pretraining Approach"[5] and was developed by Liu et al.. Roberta is trained over 160 GB of uncompressed text: Book Corpus + Wikipedia (16GB), CC-News (76GB), Open Web Text (38GB), and Stories (31GB). This model trains on larger batches and learning rates, removes the next sentence prediction objective, uses a byte-pair-encoding scheme with 50k subword units in its vocabulary (BERT uses 30k), and also works with dynamic mask generation for pretraining.

3 Methods

3.1 Data

The data set "Squad V1" was used to train the networks for each model. This data set is a reading comprehension data set, which consists of text and related questions with the relevant passages in the text meaningful for the question [7]. Many architectures are based on this data set, for example, Roberta was trained on an updated version of this data set which also consists of unanswerable questions [5]. Squad V1 is also commonly used as a benchmark tool for new natural language processing models and their evaluation and additionally, it contains multiple questions for one text [7].

3.2 Question Collection

For the evaluation of our models, we selected 15 texts from the Internet with a wide variety of topics (see figure 2). The goal was to see how well those architectures perform on arbitrarily selected texts and questions

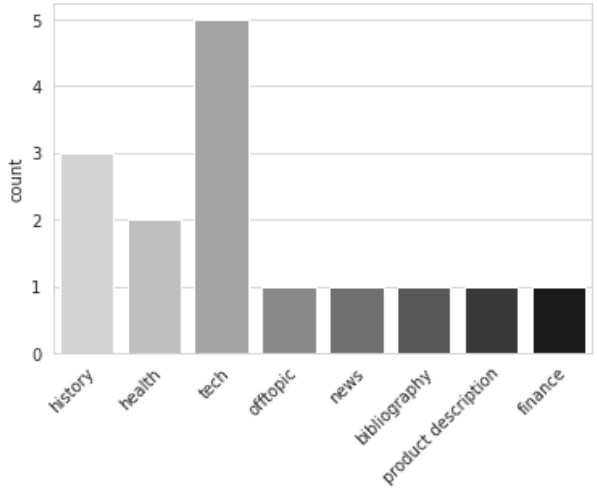

Figure 2: Barplot of the topics from the texts

Figure 2 shows the present text counts in the question collection for several different topics. One can see that most of the text correspond with the topics tech and history. All texts were then rated with a difficulty level (see figure 3). The text difficulties relate to the content of a text as well as the questions to the text.

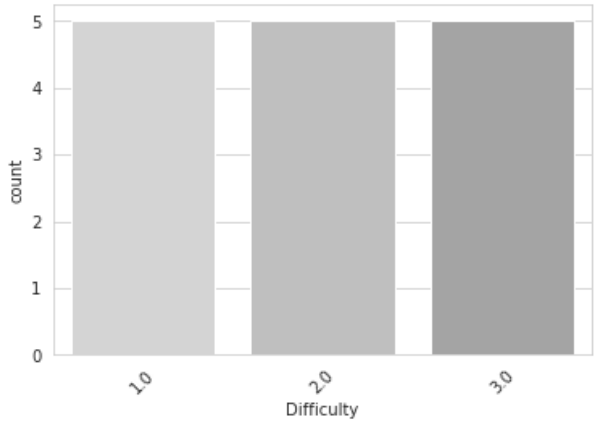

Figure 3: Barplot with count of difficulties

As visible in figure 3, there is an uniform distribution of difficulties over all topics. The difficulties are subjective and based on our opinion about the text and the related questions to the text.

3.3 Model Training

The models were created with the help of a notebook provided by Huggingface. We ran the three different models through the notebook and saved the models on Huggingface. To compare the models, the same hyper-parameters were used. The parameters were as follows:

- learning_rate: 2e-05
- train_batch_size: 16
- eval_batch_size: 16/32
- seed: 42 (for reproducibility)
- optimizer: Adam with betas=(0.9,0.999) and epsilon=1e-08
- lr_scheduler_type: linear
- num_epochs: 3

The parameters were taken directly from the notebook. From there, we could also read out the time, training loss and validation loss over epochs.

Model	Loss	Training Time (h)
Albert	0.9901	2:49:20
Distilbert	1.1460	2:32:46
Roberta	0.8953	2:37:31

Table 1: Loss ant time after training of all networks

In the table 1 one can see that the models have large deviations in loss function (Validation Loss). Roberta has the lowest loss of 0.89, more than 10% lower than Albert and up to 20% lower than Distilbert. In terms of the loss, it strongly depends on the parameters used for training for instance the learning rate, learning rate scheduler etc. Each hyper-parameter for training needs to be adapted specifically on each architecture, therefore a comparison does not make sense though. The models were trained locally with an NVIDIA GeForce RTX 3060. The time to train were rather similar for all with a batch size of 16. The fastest model was Distilbert with about 2.5 hours. As mentioned in the earlier context this model is also the lightest version and therefore it makes sense that the training was the fastest. Interesting about the graphic is that Albert needed a longer training time although it is also a lightweight version of BERT compared to Roberta which essentially consists of a larger vocabulary and should imply larger storage requirements to fulfill the training. Distilbert was the only model where the batch size was set to 32 without overloading the dedicated memory usage of the GPU. So Distilbert has a clear advantage in terms of speed.

3.4 Evaluation Metric

To evaluate and compare the implemented models, the F-Score was taken. The F-Score combines the two metrics Precision and Recall together and results in a good score if both of them are high, as it represents the harmonic mean of both [6, 31:15].

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \tag{1}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2}$$

The scores were calculated (equations 1, 2) by looking at the tokenized unigram answers and the tokenized unigram predicted answers and compare them with each other. One could also take the complete answers and compare the true and the predicted answer as an exact match. The decision to take the tokenized answers was made because it takes the partial correctness of an answer into account. All scores were averaged over all words by using macro-averaging, which simply is the mean over all words. The equal metric was also declared as reasonable for span-based prediction approaches: " [...] F1 has been generally reserved for evaluating span-based question answering. It is computed over tokens in the candidate and reference."[2]

4 Analysis

This section covers topics according to the evaluation of the trained model architectures for question answering. These topics include the analysis of the intentional trick questions according to the context, analysis of different levels of difficulties in our texts.

To start off the error analysis, the different levels of difficulties were investigated. It was hypothesised that the probabilities of the models for a predicted answer A given the context C will be high $P(C|A)$ with a decreasing level of difficulty. We created a KDE-Plot showing the distribution of the predictions of all the models to investigate this hypothesis (figure 4).

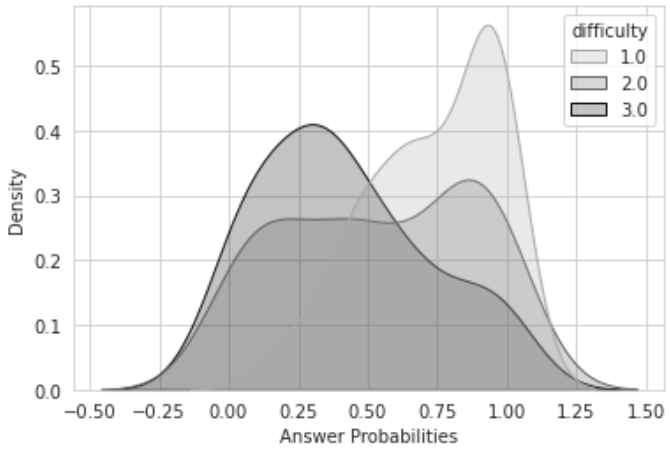

Figure 4: Comparisson of probabilities of given difficulties of text and related answers $P(C|A)$

Figure 4 shows that our questions were labelled according to the model output probabilities. This figure shows that especially the texts and questions with a difficulty level of 3 have lower probability outputs as the mode of the distribution is around 0.25. Further we can see that particularly for this difficulty level the probabilities of most answers were high. Most of the answer probabilities were around 0.9 and 1. For the middle level difficulty (2) the distribution looks more flat than the others. Particularly the upper bound of the distribution where the mode of the distribution appears (around 0.8) the density is on a much lower level compared to the density of the easiest level of difficulty (1).

Further analysis was applied to the probability values of all answers and the affect on the output performance of the model. Therefore, a scatter plot showing the relationship between both variables was made.

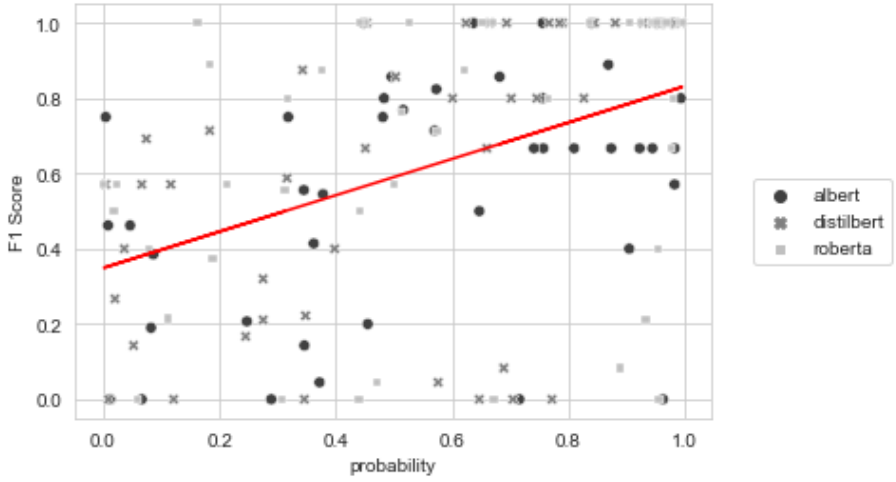

Figure 5: Scatter-plot compare probability and F1-Score with regression line (red)

Figure 5 shows a general slight increase of F1-Score if the probability increases (red line). Though, most of the dots were spread randomly accross the raster and one could not surely conclude that there is a strong relation between both variables. It is observable that we have many points showing high probability (>0.6) with low F1-score of 0 (no match) as well as points showing high probability (>0.6) with a very good F1-score of 1 (exact match).

4.1 Errors

4.1.1 Unanswerable Questions

In our 45 questions we have also asked trick questions, which are questions that are unanswerable with the given text. We wanted to know how the model behaves. Figure 6 shows a violin-plot per model with the respective probability distribution of the given best answers to a question. The violins are divided into separate classes, whereas one represents all trick questions (dark grey shade) and all non-trick questions (bright dark shade).

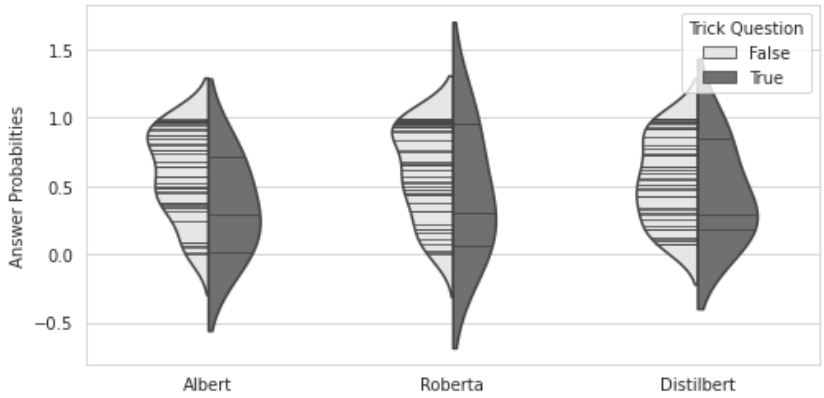

Figure 6: Comparisson of probabilities of normal questions and trick questions, which are not answerable with given context

One could expect to see low probabilities for all 3 trick questions, but this is not the case according to figure 6. 2 out of 3 trick question have low probabilities and one has high probability. The one with the high probability is one of the text and questions labelled with the highest difficulty in the collection of questions:

"When receives the Pfizer-BioNTech COVID19 vaccine full approval for children?"

This question corresponds strongly with the given context, so the model probably interprets the starting section where the context of approvals for people of certain ages and a data appears. The appearing data then results as the prediction to the question. This observation shows that the models can not interpret if a question is answerable given a matching context to the question as looks on similar words which is typical for the behaviour of these systems [6].

In text number 15 *"Pfizer-BioNTech was the first COVID-19 vaccine to receive full Food and Drug Administration (FDA) approval for people ages 16 and older in August 2021"*,

The correct answer should be "none" because the text clearly specifies that the approval was for people over 16 years of age and thus the question not answerable given the context. Both models answered August 2021 with a probability greater than 70%, being the highest Roberta (96%) [6].

As it has already been observed in figure 6, one could not just circumvent the problem when a probability threshold is set for answers, because the models tend to answer with a high probability given a similar but answerable question to the context.

To circumvent this kind of problem, which is a major task in predicting the Squad V2 data set correctly, an unanswerable question needs to be identified. The authors of one of the top competing architectures "Retrospective Reader" for the Stanford Squad V2 leaderboard state that: "When unanswerable questions are involved in the MRC [Machine Reading Comprehension] task, an essential verification module called verifier is especially required in addition to the encoder [...]"[13].

Figure 6 shows that many of the non-trick questions have a low probability associated to the given answers. One can assume that these answers could be potentially incorrect because the probability is low and so the counter probability of the model high. Figure 7 shows the output probabilities associated with right and wrong answers.

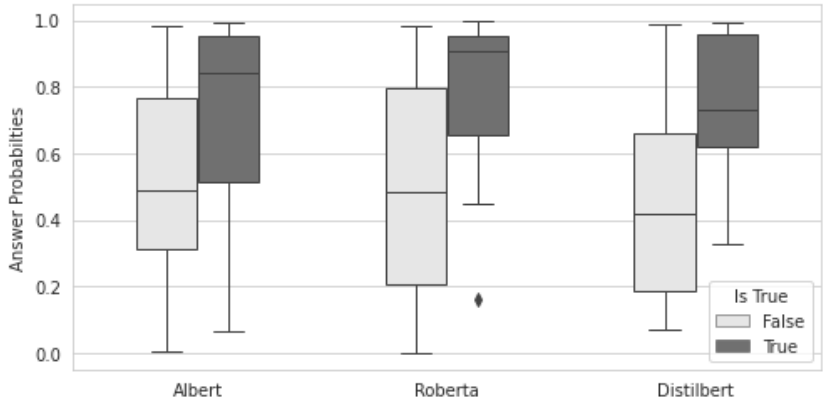

Figure 7: Comparisson of probabilities over all models of questions which were answered correct and wrong

The figure 7 shows the distribution in shape of a boxplot for each model as well as coloured in bright and dark grey if the answers were true or false. An answer was labelled as true if an exact match (EM) appeared between the untokenized answer strings. This exact match implied a resulting F1-score of 1 for an answer compared against the true answer. Figure 7 shows that all models tended to predict a signifcantly lower output probability regarding to wrong answers than for right answers. This significance is primarily high for the models Roberta and Distilbert.

4.1.2 Answer by Enumeration

The next error, which was analysed, comes at the text passage from an article about measurements of grain sizes:

Context:

"According to the measurements, data exists of three measurement systems: Miniplate Accelorometer (MPA), Square Pipe System (SPG) and Swiss Plate Geophones (SPS)."

The question that goes with this context is:

"What are the names of the three measuring systems?".

The answer here is quite clear for a human that it is the three measurement systems in the text. The models have responded as follows:

Model	Answer
ALBERT	Miniplate Accelorometer (MPA), Square Pipe System
DISTILBERT	Miniplate Accelorometer (MPA), Square Pipe System
ROBERTA	Miniplate Accelorometer (MPA),

Table 2: Answer of the models to the question *"What are the names of the three measuring systems?"*

If we look at the table 2, we see that Albert and Distilbert gave exactly the same answer. Sadly, both recognized only two measurement systems. However, they were better than Roberta. This model unfortunately has only one measurement system listed.

An clear answer of how to fix this answering issue in our models could not be found in literature. It is assumed that one could re-train the model with new parameters and check how well the re-trained model performs at this kind of question tasks. Another way was proposed by Barz et al. [1], where they have used a machine learning model to re-rank the given top answers of the implemented model and improved the results in their application [1]. This method though implies that one of the top answers needs to include the correct answer, otherwise this method can not be successfully applied on our use-case. Regarding to this it was tested on the Distilbert model which top answers were returned on the given question and context from above (see figure 10). All measuring systems were listed but not in the same answer span, which would make it difficult to apply a post-processing step to increase the answer performance with this method.

An alternative approach for a solution to this enumeration problem was the paper on Multi-span Questions [9]. The focus of this paper is that generally models for reading comprehension focus on only one contiguous section to alleviate learning problems but can be restrictive when an answer consists of a number of non-contiguous sections in the text. While our answer actually consists of one contiguous section for humans, all three models, which were trained, struggled to list all measurement systems. Here, perhaps the multi-span question could come to a correct solution, because we have seen from the analysis of the top answers (see figure 10) that it finds all the measurement systems, but never lists them at the same time. In the approach of multi-span answers, they propose: " [...] a simple architecture for answering multi-span questions by casting the task as a sequence tagging problem, namely, predicting for each input token whether it should be part of the output or not." [9].

4.1.3 Context

A third type of error was found in the following text:

"In 2019, the social network company Facebook launched a social VR world called Facebook Horizon. In 2021 Facebook was renamed "Meta Platforms" and its chairman Mark Zuckerberg declared a company committed to developing a metaverse."

where the corresponding question was:

"Do you know of any companies that are committed to developing the metaverse concept?"

Although at the beginning of the text the word Facebook appears related to the word company ("The social network company Facebook..."), only the Roberta-model was able to answer correctly, although with a low probability (0.16).

It is even more difficult to explain, as the other two models answered to the question Mark Zuckerberg, although Albert with a very low probability (0.06), being specified in the text that this person is the chairman of the aforementioned company ("...and its chairman Mark Zuckerberg declared a company committed to developing a metaverse").

Although both models were trained with the same hyperparameters, it is hard to answer exactly why Roberta answers the question correctly, while Albert and Distilbert do not.

A first approach to the problem could be to reformulate the question, for example, which company invests in the metaverse? Or which company is Mark Zuckerberg the president of?

However, understanding the differences between both models in their pretraining stage can give us a clue in the search for one or more causes.

What actually makes Roberta perform better than other models, or even better than the BERT model itself in question and answer projects is: Not only was it trained with 10 times more data and a larger batch size on longer sequences, but also the process of tokenization is different, it uses a larger byte-level Byte Pair Encoding (BPE), which means, a hybrid between character and word-level representations, that allows handling the large vocabularies common in natural language processing corpora. Instead of full words, Byte Pair Encoding relies on sub-words units, which are extracted by performing statistical analysis of the training corpus [10]. BPE ensures that the most common words are represented in the vocabulary as a single token, while the rare words are broken down into two or more sub-word tokens.

At the same time, instead of using the next sentence prediction NSP, Roberta works with full sentence, that means, each input is packed with full sentences sampled contiguously from one or more documents, such that the total length is at most 512 tokens. Inputs may cross document boundaries. When we reach the end of one document, we begin sampling sentences from the next document and add an extra separator token between documents. We remove the NSP loss [5].

While Albert uses the sentence order prediction (SOP) approach. It consists in that Albert takes two consecutive segments from the same document as a positive class, and then swap the order of the same segment and use that as a negative example. The efficiency over Next Sentence Prediction (NSP) from BERT is that NPS actually learns whether the two sentences belong to the same topic, which is much easier than learning whether the sentences are grammatically coherent or not.

Although there are other differences between both models, we mainly rescue these, as causes of Roberta's better performance when answering questions given a context.

4.2 Model Comparisson

For the evaluation and comparison of all models, barplots (figure 8) were created for each model and question difficulty.

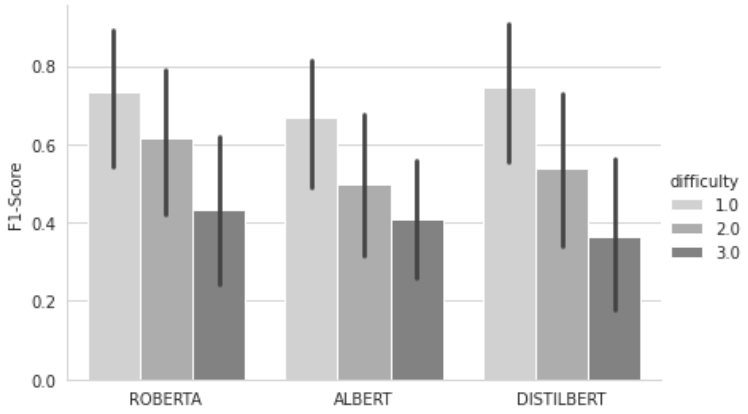

Figure 8: Evaluation of F1-scores grouped by model and question difficulties, black candle indicates the 95% confidence interval

Although the difficulty labeling for each text was done subjectively by ourselves, the graph above shows that in fact the classification is correct. In each of the models, the best score was obtained for those questions with low difficulty, and the score was reduced as the difficulty increased.

The level of difficulty also showed some advantages in each model. For example, when the difficulty level is one, all models perform very similarly and well, but at difficulty level two or three, Roberta scores better than the remaining two networks.

One of the other observations, which could be made with the question examples, was that Albert was the only model that always took the one punctuation mark after the correct answer. This was of course unfavourable for the scoring in the evaluation. It was a drawback for this model within the scoring process, which distorts the output score.

5 Conclusion

At the end, we recognized that our models were trained on the Squad V1 data set and not on the Squad V2 data set. It would be interesting to see how well models trained on the Squad V2 data set perform on unanswerable questions and what their answers were, as the Squad V2 data set contains many of unanswerable questions. Furthermore, a comparison of the fitted models and the leaderboard models could be interesting.

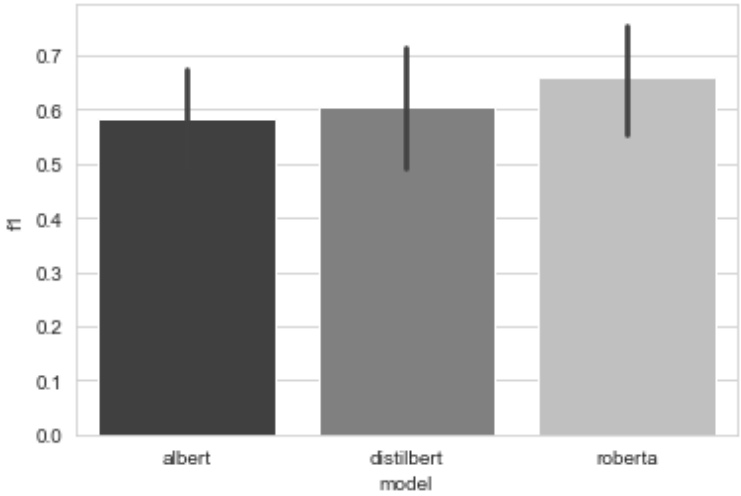

Figure 9: Barplot of the Models compare

In summary, we can see from the barplot of the figure 11 that Roberta performs best with a small lead and has a F1 score of about 0.65. Albert, on the other hand, is the worst model, but this may also be because of the punctuation marks. With the removal of the punctuation marks, Albert will certainly become a strong competitor to the other models. In addition, Roberta also had the deepest training and validation loss, which has actually already been an indication for us that Roberta should perform better. Apart from the performance, we also looked at the time of the training, because it went on for a very long time. The shortest time for training needed Distilbert with two and a half hours with 3 epochs and a batch size of 16. Albert had the longest time.

References

[1] M. Barz and D. Sonntag. Incremental Improvement of a Question Answering System by Re-ranking Answer Candidates using Machine Learning. *arXiv:1908.10149 [cs, stat]*, 714:367–379, 2021. arXiv: 1908.10149.

[2] A. Chen, G. Stanovsky, S. Singh, and M. Gardner. Evaluating Question Answering Evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

[4] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv:1909.11942 [cs]*, Feb. 2020. arXiv: 1909.11942.

[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, July 2019. arXiv: 1907.11692.

[6] Manning. Stanford CS224N: NLP with Deep Learning | Winter 2019 | Lecture 10 – Question Answering, Mar. 2019.

[7] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv:1606.05250 [cs]*, Oct. 2016. arXiv: 1606.05250.

[8] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*, Feb. 2020. arXiv: 1910.01108.

[9] E. Segal, A. Efrat, M. Shoham, A. Globerson, and J. Berant. A Simple and Effective Model for Answering Multi-span Questions. *arXiv:1909.13375 [cs]*, Oct. 2020. arXiv: 1909.13375.

[10] R. Sennrich, B. Haddow, and A. Birch. *Neural machine translation of rare words with subword units.* In Association for Computational Linguistics (ACL). 2016.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, Dec. 2017. arXiv: 1706.03762.

[12] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*, July 2020. arXiv: 1910.03771.

[13] Z. Zhang, J. Yang, and H. Zhao. Retrospective Reader for Machine Reading Comprehension. *arXiv:2001.09694 [cs]*, Dec. 2020. arXiv: 2001.09694.

A Tables

Training Loss	Epoch	Step	Validation Loss
0.8584	1.0	5540	0.9056
0.6473	2.0	11080	08975
0.4801	3.0	16620	0.9901

Table 3: Training Loss and Validation Loss of Albert

Training Loss	Epoch	Step	Validation Loss
1.2856	1.0	2767	1.1919
1.012	2.0	5534	1.1332
0.8512	3.0	8301	1.1460

Table 4: Training Loss and Validation Loss of Distilbert

Training Loss	Epoch	Step	Validation Loss
0.8926	1.0	5536	0.8694
0.6821	2.0	11072	0.8428
0.5335	3.0	16608	0.8953

Table 5: Training Loss and Validation Loss of Roberta

B Figures

```
[60]: {'question': 'What are the names of the three measuring systems?',
       'answer': [{'score': 0.07973705977201462,
         'start': 383,
         'end': 432,
         'answer': 'Miniplate Accelorometer (MPA), Square Pipe System'},
        {'score': 0.04566909745335579,
         'start': 383,
         'end': 412,
         'answer': 'Miniplate Accelorometer (MPA)'},
        {'score': 0.031063061207532883,
         'start': 383,
         'end': 406,
         'answer': 'Miniplate Accelorometer'},
        {'score': 0.03046952188014984,
         'start': 443,
         'end': 470,
         'answer': 'Swiss Plate Geophones (SPS)'},
        {'score': 0.0032222166191786528,
         'start': 443,
         'end': 464,
         'answer': 'Swiss Plate Geophones'},
```

Figure 10: Snippet of Answers in a listed context

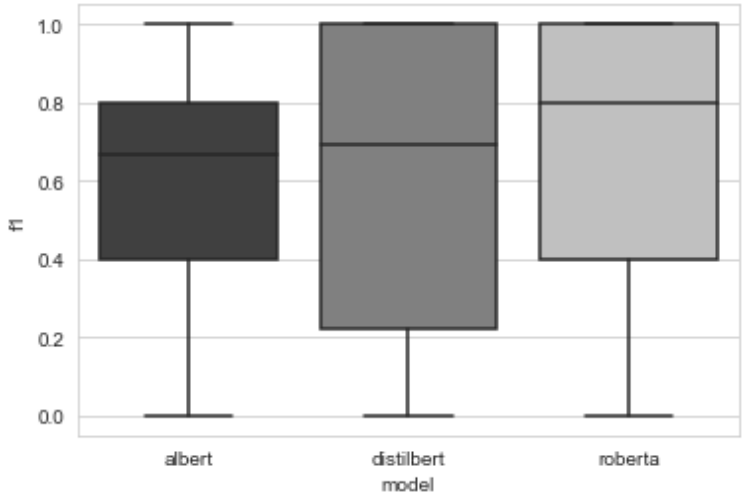

Figure 11: Boxplot of the Models compare