npr: Mini-Challenge 2, Question & Answering

Roman Studer, Vincenzo Timmel, Lukas Gehrig

Bachelor Data Science (cand.)

{roman.studer1, vincenzo.timmel, lukas.gehrig}@students.fhnw.ch

FHNW, Brugg-Windisch

npr: Mini-Challenge 2, Question & Answering

Introduction

Question & Answering tasks are increasingly solved by transformer models, such as BERT. This form of transfer learning is becoming increasingly popular. The introduction of large data sets for specific NLP tasks has provided a leap in the performance of transformers. For example, the SQuAD (The Stanford Question Answering Dataset) dataset of 100,000 questions has achieved performance above human ability on models such as IE-NET, Albert, FPNET, etc [1]. We now consider the performance of several models on the SQuAD v2 dataset and discuss some possible sources of error.

Background

Dataset

The used dataset SQuAD v2 is the continuation of the well-known dataset SQuAD. This consists of questions asked on a set of Wikipedia articles. The answers are excerpts from these articles which are given in the Q&A problem as context for the model. In addition to SQuAD, SQuAD v2 also has questions that do not have a matching answer in context. Thus, the model must also be able to learn when there is no answer in the context [1], [2]. Models that performed superhumanly on SQuAD are not guaranteed to achieve this on SQuAD v2. This is because these models simply select the part of the context that has the greatest relationship to the question and do not check whether the answer actually answers the question.

With SQuAD v2, the model must additionally learn whether the part selected as the answer also contains a plausible answer. [2].

Previous Works

With the introduction of transformer models such as BERT, the human-level performance threshold was surpassed shortly after the introduction of the SQuAD v2 dataset. As of 29/01/2022, an IE-Net (Ensemble) model holds the record with a score of 90.939 EM. For comparison, Human Level Performance achieves a score of 86,831 EM [1], [3].

Model Architecture

Since the introduction of transformer models, several variations and reinterpretations have been established. We have trained and evaluated a total of three models. We use two BERT-based models, DistilBERT and ELECTRA, as well as the XLNET model, which uses ideas from Transformer-XL, an autoregressive model [4].

DistilBERT

DistilBERT is a BERT-based transformer model that achieves 97% of the performance of large BERT models with only 40% of the size of BERT. This allows for fast and resource-efficient finetuning, making the application of the model more accessible [5]. Using the idea of "knowledge distillation", where a smaller model mimics the behaviour of a larger one, DistilBERT can excel at downstream tasks such as Question Answering. On the SQuAD dataset, DistilBERT is 3.9 points behind BERT [5].

ELECTRA

ELECTRA's underlying model is BERT. BERT masks pre-training tokens with [MASK] in masked language modelling (MLM). Kevin Clark et al. describe an alternative approach in their paper on ELECTRA [6]. Masking tokens achieves good results on downstream tasks but requires a lot of computational power to be effective. ELECTRA replaces the words to be masked with plausible alternatives using a generator network. A model was then trained on this adjusted data to identify which tokens were replaced. As a result, ELECTRA achieves better contextual representations than BERT [6].

XLNET

XLNET integrates ideas from the Transformer-XL model, an autoregressive model, in pretraining to apply a generalized autoregressive pretraining method [4]. XLNET makes all hidden states available to the encoder. It thus maps input directly to output.

Metrics

The performance of the trained models is considered using two metrics in this report. We use both F1, a metric that is also included in the official evaluation script of SQuAD v2 [1], and BLEU, a widely used metric for rapid scoring of machine translation models [7]. However, BLEU can also be used for Question & Answering.

F1-Score

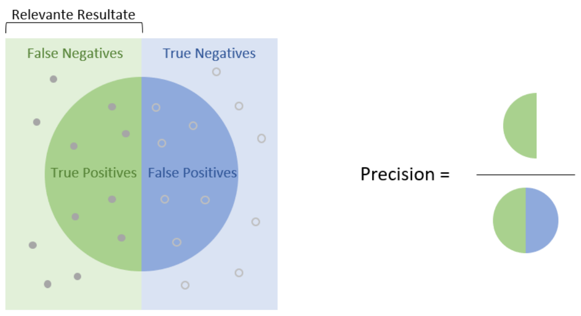


Figure 1, Precision

F1 is formed from the harmonic mean of Precision and Recall. Like an IR task, Precision is the ratio of the number of shared worlds to the total number of words in the prediction and recall is the ratio of shared words to the total number of words in the ground truth [8] (see Figure 1). The formula for F1 is given with Precision and Recall as follows:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (1)$$

A property of the harmonic mean is that F1 can only be high if both Precision and Recall are simultaneously high. In the case of Question & Answering on the SQuAD v2 dataset, this means that a high F1 score is only possible if the answer contains few words in the answer that are not in the reference and that the answer contains about the same number of words as the references. The special case that no answer exists needs special consideration. If the model gives no answer even though a reference answer exists, it receives an F1 score of 0 for this question, which is a false positive. The same applies to the false negative samples. However, if the model does not give an answer and the reference answer does not exist, it is a true positive and then the model receives the F1 score 1 for this question [1].

BLEU-Score

Papineni et al. launched the Bilingual Evaluation Understudy (BLEU) score for easier and faster evaluation of translation models [7]. The metric can also be used for Question & Answering. In both use cases, an answer of the model is available which is compared to a set of references. In the case of Q&A with SQuAD v2, a part of the context and three answers as references. With a modified N-Gram precision and a regularization factor (BP, Brevity Penalty) the BLEU score is calculated. The formula for the calculation is

$$BLEU = BP \cdot exp\left(\sum_{\{n=1\}}^{N} w_n log p_n\right) \tag{2}$$

, where p_n is the N-gram precision score and $w_n = 1/N$ is the weight. The N-gram length is denoted by N [7]. The regularization factor BP is calculated from the length c of the generated response and r, the length of the reference, as follows [7]:

$$BP = \begin{cases} 1 if c > r \\ e^{\left(1-\frac{r}{c}\right)} if c \leq r \end{cases} \tag{3}$$

However, since in SQuAD v2 no answer is a possibility, we implemented an adjusted BLEU score which punishes, respectively rewards cases when no answer in either or both the prediction and the ground truth appears. This is done by setting the score at 1 if both the prediction and the ground truth have no answer or at 0 if the model gives no answer when an answer is expected or answers if no answer would be correct.

Experiments

Training

All three models were trained, respectively validated, with the same train and validation set. The dataset was loaded using Huggingface. This library provides the dataset as an object, already divided into train and validation set. The models were trained locally on a GPU with three epochs each. A learning rate of 2e-5 was used.

Training DistilBERT

The parameters for the DistilBERT model were set as follows. Note that its larger batch size leads to better results. Albeit with decreasing yield gains. In addition, the hardware used must be able to work with a larger batch size. DistilBERT allows a maximum number of tokens of 512. The parameters used are:

```
BATCH_SIZE: 16
MAX_LENGTH: 384
DOC_STRIDE: 128
N_BEST_SIZE: 20
MAX_ANSWER_LENGTH: 30
```

Training ELECTRA

Since Electra is only an alternative pretraining method for BERT, the same limitations apply [9]. Therefore, the parameters were again set as follows:

```
BATCH_SIZE: 16
MAX_LENGTH: 384
DOC_STRIDE: 128
N_BEST_SIZE: 20
MAX_ANSWER_LENGTH: 30
```

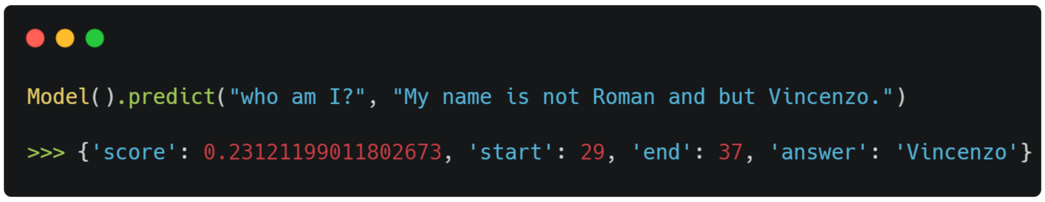
Training XLNET

Compared to BERT, XLNET has no limitation on the number of tokens [4]. Due to the hardware used for training, the MAX_LENGTH had to be reduced in this case.

```
BATCH_SIZE: 8
MAX_LENGTH: 300
DOC_STRIDE: 100 # 1/3 MAX_LENGTH
N_BEST_SIZE: 20
MAX_ANSWER_LENGTH: 30
```

Prediction

After training, the models were stored locally for later analysis. Prediction is now possible. Via a Question Answering Pipeline, a question can now be answered on a context text. Example:



```
Model().predict("who am I?", "My name is not Roman and but Vincenzo.")
>>> {'score': 0.23121199011802673, 'start': 29, 'end': 37, 'answer': 'Vincenzo'}
```

Figure 2, Example of a predicted answer for a Q&A-Model

Figure 2 shows how the pipeline can be used. A question, in the form of a string, is given together with a context text, also directly as a string to the predict function. This function executes tokenizer and model in the background. The answer is a dictionary with the score (confidence to the answer), the start and end index, as well as the answer transformed back into text. The answer is a section of the context.

Results

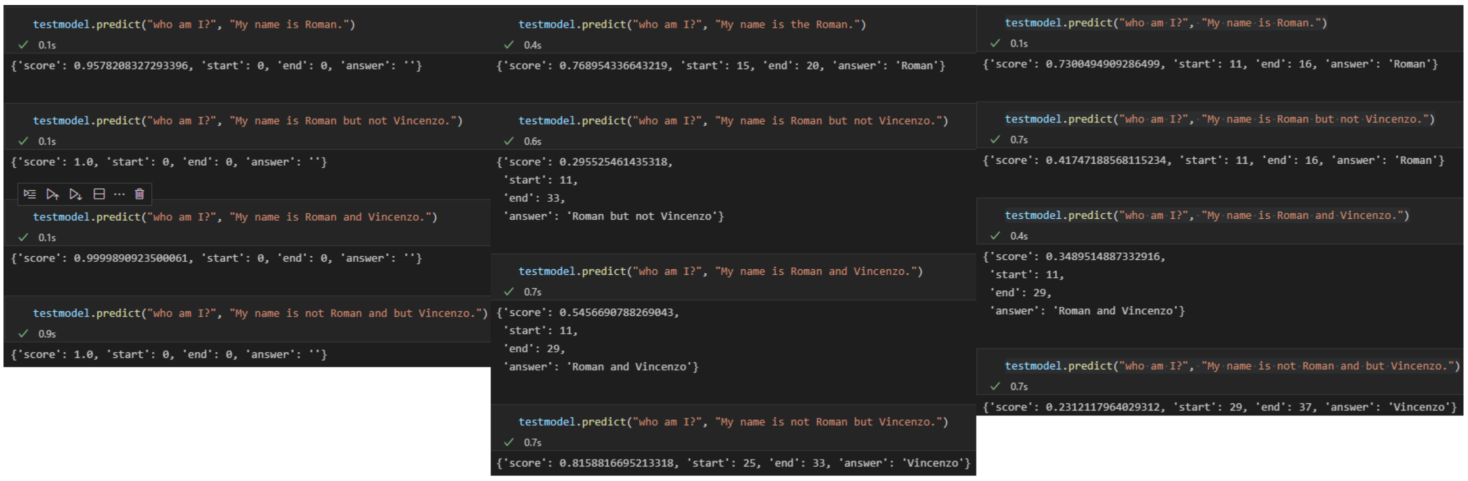
In Table 1, we compare the trained models using the introduced metrics, BLEU and F1. BLEU in our case only considers the results where an answer prediction was made, and the ground truth also contained an answer. So, we can evaluate how good the answers actually are if one exists. In addition, adj. BLEU considers not only the true positive answers, but also all the other predictions as described in BLEU-score. An N-gram of length 2 was chosen to calculate BLEU, because there are a lot of answers with just a few words in the dataset.

Table 1, Comparison of the models according to Exact Match, BLEU, adj. BLEU and F1

Model	Exact	BLEU-2	Adj. BLEU-2	F1	F1-HasAns	F1-NoAns
DistilBERT	59.563	0.7996	0.644	0.6518	0.6690	0.6345
ELECTRA	63.067	0.7979	0.676	0.6822	0.6615	0.7031
XLNET	55.394	0.8012	0.567	0.5691	0.2027	0.9344

Already with a few epochs, all models achieved a BLEU score of about 0.8. XLNET reached the best feasible answers to the questions if a prediction was made due to its high BLEU-score of 0.8012. However, it is the worst considering the F1 score. The results confirm the data published in the respective papers about the performance of the models. Thus, we can confirm that XLNET can outperform BERT models on the Question Answering task [4]. We see a high difference between the F1-HasAns (A response exists), as well as F1-NoAns (No response exists) for XLNET. The model seems to predict no answer very often. With about 50% unanswerable questions in the training set, this allows for a high F1 score on this subset. A high F1 score is also achieved because most no answer questions, are answered correctly. This issue of carelessly not giving an answer seems to be an existing problem in the current implementation of XLNET on Huggingface, as confirmed by the following issue on GitHub: XLNET SQuAD2.0 Fine-Tuning - What May Have Changed? This was detected by us but not solved for this mini-challenge.

This can also be seen in our small test, in which we wanted to test with simple cases if the model can understand slight differences between questions.



```
testmodel.predict("who am I?", "My name is Roman.")
✓ 0.1s
{'score': 0.9578208327293396, 'start': 0, 'end': 0, 'answer': ''}

testmodel.predict("who am I?", "My name is Roman but not Vincenzo.")
✓ 0.1s
{'score': 1.0, 'start': 0, 'end': 0, 'answer': ''}

testmodel.predict("who am I?", "My name is Roman and Vincenzo.")
✓ 0.1s
{'score': 0.9999890923500061, 'start': 0, 'end': 0, 'answer': ''}

testmodel.predict("who am I?", "My name is not Roman and but Vincenzo.")
✓ 0.9s
{'score': 1.0, 'start': 0, 'end': 0, 'answer': ''}
```

```
testmodel.predict("who am I?", "My name is the Roman.")
✓ 0.4s
{'score': 0.768954336643219, 'start': 15, 'end': 20, 'answer': 'Roman'}

testmodel.predict("who am I?", "My name is Roman but not Vincenzo.")
✓ 0.6s
{'score': 0.295525461435318,
 'start': 11,
 'end': 33,
 'answer': 'Roman but not Vincenzo'}

testmodel.predict("who am I?", "My name is Roman and Vincenzo.")
✓ 0.7s
{'score': 0.5456690788269043,
 'start': 11,
 'end': 29,
 'answer': 'Roman and Vincenzo'}

testmodel.predict("who am I?", "My name is not Roman but Vincenzo.")
✓ 0.7s
{'score': 0.8158816695213318, 'start': 25, 'end': 33, 'answer': 'Vincenzo'}
```

```
testmodel.predict("who am I?", "My name is Roman.")
✓ 0.1s
{'score': 0.7300494909286499, 'start': 11, 'end': 16, 'answer': 'Roman'}

testmodel.predict("who am I?", "My name is Roman but not Vincenzo.")
✓ 0.7s
{'score': 0.41747188568115234, 'start': 11, 'end': 16, 'answer': 'Roman'}

testmodel.predict("who am I?", "My name is Roman and Vincenzo.")
✓ 0.4s
{'score': 0.3489514887332916,
 'start': 11,
 'end': 29,
 'answer': 'Roman and Vincenzo'}

testmodel.predict("who am I?", "My name is not Roman and but Vincenzo.")
✓ 0.7s
{'score': 0.2312117964029312, 'start': 29, 'end': 37, 'answer': 'Vincenzo'}
```

Figure 3, Sample predictions of XLNet, ELECTRA and DistilBERT. XLNet always returns no answer. DistilBERT and ELECTRA both answers questions well but DistilBERT also correctly interprets the context in the second questions by answering "Roman" instead of "Roman but not Vincenzo".

The two BERT models, DistilBERT and ELECTRA have similar F1 values for both classes (response possible and no response possible). The problems encountered in XLNET do not exist in DistilBERT and ELECTRA.

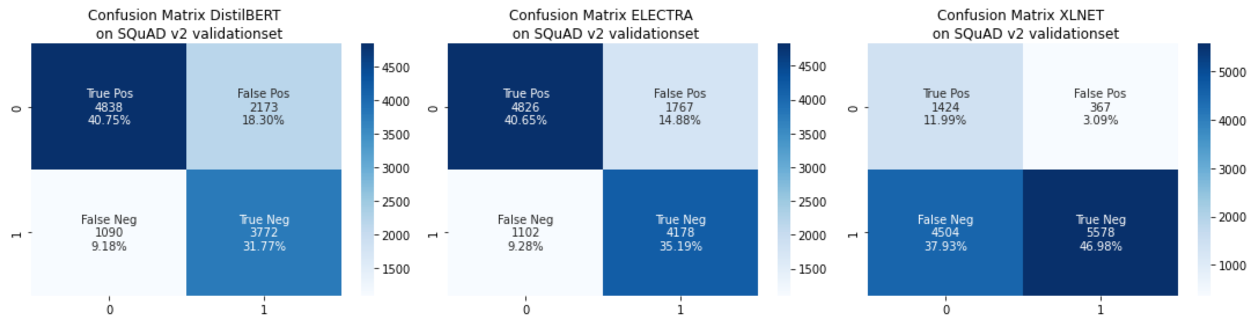


Figure 4, Confusion matrices of all three models

Figure 4 shows the Confusion Matrix and metrics calculated on the Confusion Matrix across all three models. The True Negative, True Positive, False Negative, and False Negative classes are defined as follows: All four metrics are based on the length of the output and the length of the reference. True Positive occurs when both response and reference have the same length, True Negative when both response and reference have length zero, False Positive when a response exists and no reference, and False Negative when a reference exists but no response was given. Figure 4 shows that DistilBERT produces an answer in just under 20% of the cases although none exists and with just over 30% correctly no answer is produced. At 40%, the correct answer is produced when one also exists. This corresponds to an accuracy of about 55% in the subset of questions that have a reference. A very similar picture emerges for the ELECTRA model. As already seen in Table 1, we get many false negatives with XLNET. They account for about 38% of the answers. Often not giving an answer leads to a high True Negative rate, since 50% of the test data set consists of questions for which there is no answer in the context.

Error Analysis

As written before in **Results**, we had problem with the fine-tuning of XLNet because the model often gave no answer when an answer is expected. This error could probably be addressed by fixing the code. Because other people have a similar problem, we suspect that the error is in the package and not an error caused by wrong usage of the model, especially when the other two models work flawlessly.

Sometimes, models mistook similar things for the same thing (e.g. ELECTRA saying that *president* is a *premier minister*, see electra_prediction.ipynb) and thus answering a question, which cannot be answered. This could be improved by using a more advanced model. Another error which could be fixed the same way are when questions are asked in a philosophical way (e.g. the question "*When did violence end in war", see* electra_prediction.ipynb*)*, which either have no clear answer or many answers are correct. Those should also be detected by the model and the model should avoid answering them. The mistakes can be seen when looking at the *Confusion Matrix* and their examples in the evaluation notebooks (see Figure 4).

Other questions are also written in a way, that only a directional word (e.g., *into* instead of *around*, see distilbert_prediction.ipynb) changes the question from being easily answerable to not answerable and DistilBERT does this currently wrong. However, it should also be fixable by a more advanced model.

As seen in *Figure 4*, DistilBERT correctly realises that "*but not Vincenzo"* is probably not part of a name. However, ELECTRA does not. This means that a model should also not only be selected on metrics but also based on the task at hand. Thus, the best model for task A is probably not the best model for task B because in our evaluation, ELECTRA came out as being the best model.

Conclusion

In our comparison, we have shown how three transformer models behave on the SQuAD v2 data set. All models achieve a similar BLEU score of about 80%. A metric that is more meaningful compared to the F1 score, since it also considers the position of the words in the predicted response due to the n-grams. The adj. BLEU score also handles the cases with no answer. Based on our evaluation, we recommend ELECTRA because it reaches the highest F1-Score, a high BLEU score, the highest "adj. BLEU" score and the highest exact matches while being by far the fastest to train (~90 minutes with an RTX 3060Ti, compared to ~7 hours for both XLNet and DistilBERT) and being the smallest model size (~50MB, compared to 240MB for DistilBERT and 420MB for XLNet). We recommend training the models with more performant hardware on larger batch size and higher max sentence length to build on the scores obtained here.

Appendix

Please refer to the GitHub-Repository.

Sources

[1] «The Stanford Question Answering Dataset». https://rajpurkar.github.io/SQuAD-explorer/ (zugegriffen 29. Januar 2022).

[2] P. Rajpurkar, R. Jia, und P. Liang, «Know What You Don't Know: Unanswerable Questions for SQuAD», *ArXiv180603822 Cs*, Juni 2018, Zugegriffen: 29. Januar 2022. [Online]. Verfügbar unter: http://arxiv.org/abs/1806.03822

[3] «Papers with Code - SQuAD2.0 Benchmark (Question Answering)». https://paperswithcode.com/sota/question-answering-on-squad20?metric=F1 (zugegriffen 29. Januar 2022).

[4] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, und Q. V. Le, «XLNet: Generalized Autoregressive Pretraining for Language Understanding», *ArXiv190608237 Cs*, Jan. 2020, Zugegriffen: 29. Januar 2022. [Online]. Verfügbar unter: http://arxiv.org/abs/1906.08237

[5] V. Sanh, L. Debut, J. Chaumond, und T. Wolf, «DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter», *ArXiv191001108 Cs*, Feb. 2020, Zugegriffen: 29. Januar 2022. [Online]. Verfügbar unter: http://arxiv.org/abs/1910.01108

[6] K. Clark, M.-T. Luong, und Q. V. Le, «ELECTRA: PRE-TRAINING TEXT ENCODERS AS DISCRIMINATORS RATHER THAN GENERATORS», S. 18, 2020.

[7] K. Papineni, S. Roukos, T. Ward, und W.-J. Zhu, «Bleu: a Method for Automatic Evaluation of Machine Translation», in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, Juli 2002, S. 311–318. doi: 10.3115/1073083.1073135.

[8] «Evaluating QA: Metrics, Predictions, and the Null Response», *NLP for Question Answering*, 9. Juni 2020. https://qa.fastforwardlabs.com/no%20answer/null%20threshold/bert/distilbert/exact%20match/f1/robust%20predictions/2020/06/09/Evaluating_BERT_on_SQuAD.html (zugegriffen 30. Januar 2022).

[9] «ELECTRA». https://huggingface.co/docs/transformers/v4.16.1/en/model_doc/electra (zugegriffen 29. January 2022).