Proceedings of the OSACT 2022 Workshop @LREC2022, pages 126–129
Marseille, 20 June 2022

niksss at Qur'an QA 2022: A Heavily Optimized BERT Based Model for Answering Questions from the Holy Qu'ran

Nikhil Singh
nikhil3198@gmail.com

Abstract

This paper presents the system description by team niksss for the Qur'an QA 2022 Shared Task. The goal of this shared task was to evaluate systems for Arabic Reading Comprehension over the Holy Quran. The task was set up as a question-answering task, such that, given a passage from the Holy Quran (consisting of consecutive verses in a specific surah(Chapter)) and a question (posed in Modern Standard Arabic (MSA)) over that passage, the system is required to extract a span of text from that passage as an answer to the question. The span was required to be an exact sub-string of the passage. We attempted to solve this task using three techniques namely conditional text-to-text generation, embedding clustering, and transformers-based question answering. The codes for the submitted system are available at: `https://github.com/nikhilbyte/Quran-Question-Answering`

Keywords: Question Answering,Machine Reading Comprehension, Arabic

1. Introduction

The Holy Quran is the central religious text of Islam and is held by over 1.8 billion[[1]] people in 47 languages and specific verses available in 114 languages across the world. This makes it one of the most popular books in the history of mankind. This shared task (Malhas et al., 2022) addresses the challenge of answering the questions in Modern Standard Arabic by inculcating the knowledge from the verses of the Holy Quran. Moreover most of the studies dealt with factoid (what, where, who, when, which, and how much/many) questions (Abdelnasser et al., 2014),(Gusmita et al., 2014),(Hakkoum and Raghay, 2016) and a very few non-factoid questions (Hakkoum and Raghay, 2015),(Alqahtani and Atwell, 2016). This task contains questions from both factoid and non-factoid types making it a harder task.

Closed domain information retrieval pertaining to the Holy Quran has been seldom explored. Techniques such as String Matching, Probabilistic Model, and Natural Language Processing have been used in the past to extract the answers from the holy Quran. These techniques, however, give inaccurate results when the user inputs their query in natural language as these techniques retrieve the answers on the basis of keywords provided by the users. This sprouts the need for a Question Answering System(QAS) which provides slightly accurate results when compared to the techniques mentioned above.

Due to the recent advancement of the transformer architecture in QAS in multiple languages and multiple domains, we decided to experiment with transformers based models to tackle this challenge.

2. Dataset

The dataset for the task is called Qur'anic Reading Comprehension Dataset, abbreviated as QRCD[2] (Malhas and Elsayed, 2020). It is composed of 1,093 tuples of question-passage pairs that are coupled with their extracted answers to constitute 1,337 question-passage-answer triplets. Out of these 1337 triplets, 65% were allowed to train the model, 10% triplets were kept as a development set and the rest 25% triplets were reserved to evaluate the performance of the submitted system. The data distribution can be seen in Table **??**.

Dataset	#Question-Passage-Answer Triplets
Training	861
Development	128
Test	348

Table 1: Data Distribution

3. Experiments

3.1. Semantic Embeddings and Clustering(SEC)

The first experiment involved the creation of a basic sentence embedding that contained all the semantic information of a given sentence in a 786-dimensional feature vector which was generated by passing the sentence through 12 layers of transformer blocks. The transformers architecture used in all of our experiments is AraBERT(Antoun et al., 2020). The detailed steps involved in this experiment is presented below.

- Of a given Question-Answer-Passage triplet, the passage was first tokenized sentence-wise using the nltk's sent-tokenize function from the nltk library[3].

[1] ThoughtC (`https://www.thoughtco.com`)

[2] `https://gitlab.com/bigirqu/quranqa`

[3] `https://www.nltk.org/`

- Sentence embeddings were extracted from the [CLS] token from the layer of BERT model for each sentence.
- The correlation between individual sentences and the result of Principal component analysis after applying K-Means Clustering(Lloyd, 1982) is visualized. A different number of clusters were found in different passages.
- Finally the text pair of Question-Answer were passed through the same model individually and were projected onto the same Euclidean space as the tokenized passage sentences.
- This Euclidean space was then optimized using the K- Nearest Neighbors (Fix and Hodges, 1989) for bringing the question embedding closer to the cluster containing the answer sentence.
- The question and passages were passed for inference and top 5 sentences from the nearest clusters were taken as predictions.

3.2. Seq2Seq based text span extraction(S2S)

In this experiment, we treated this problem as a text2text generation problem where the input for the model model was the Question and Passage pair separated by a [SEP] token and the model was required to generate the text span from the passage as the answer.State of the art generative models such as, mT5(Xue et al., 2020) and mBART(Liu et al., 2020) were used as the Seq2Seq models for this experiment. These models performed the best for ”Exact-Match” metric out of all the experiments we conducted, however we couldn’t bring them to generate more than one text sentence as required for evaluation on this shared task.

3.3. Fine-Tuning and Optimizing BERT model

This experiment constitutes our final experiment. In this, we fine-tune a BERT-based model which takes the question-passage pair as a singe-packed sequence which is converted into the input embedding by taking a sum of the token embedding and the segment embedding to distinguish between the two as shown in Figure 1.

The fine-tuning of a BERT-based QAS involves optimization of finding the start and end word of the text span inside the passed reference text. To find the probability of a particular word being the starting and the ending word of the answer span, it takes the dot-product between the final embedding of the words in the sequence and the start/end vector, which is a 768-dimensional vector and is the same for all the words. The vector obtained after the dot product is passed to a classification layer, which outputs the probability of the word being the starting token and ending token. The complete working of our model is shown in Figure 1.

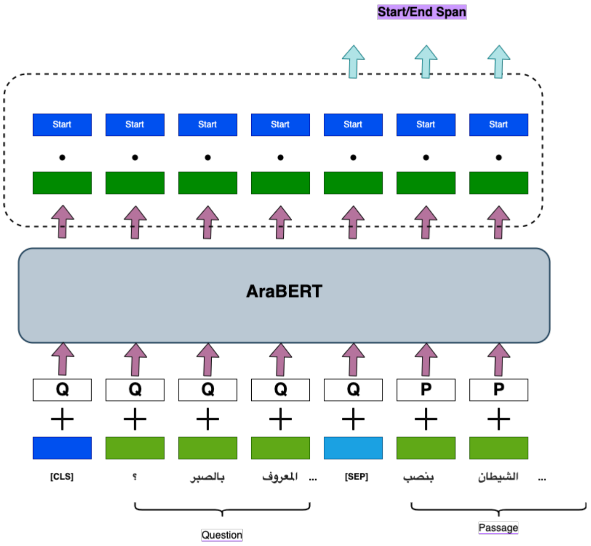


Figure 1: Model Architecture

We use Simple Transformers[4] for Conducting this Experiment.

3.3.1. Hyperparameter Optimization

We took the following Hyperparameters into consideration:

- Learning Rate
- Max Sequence Length
- Number of Epochs
- Train Batch Size

And we judged the performance of the model by the following parameters as Highlighted in Figure 2:

- Evaluation Loss
- Training Loss
- Number of Correct Answers
- Number of Incorrect answers

The results on validation set is shown in the Table 2:

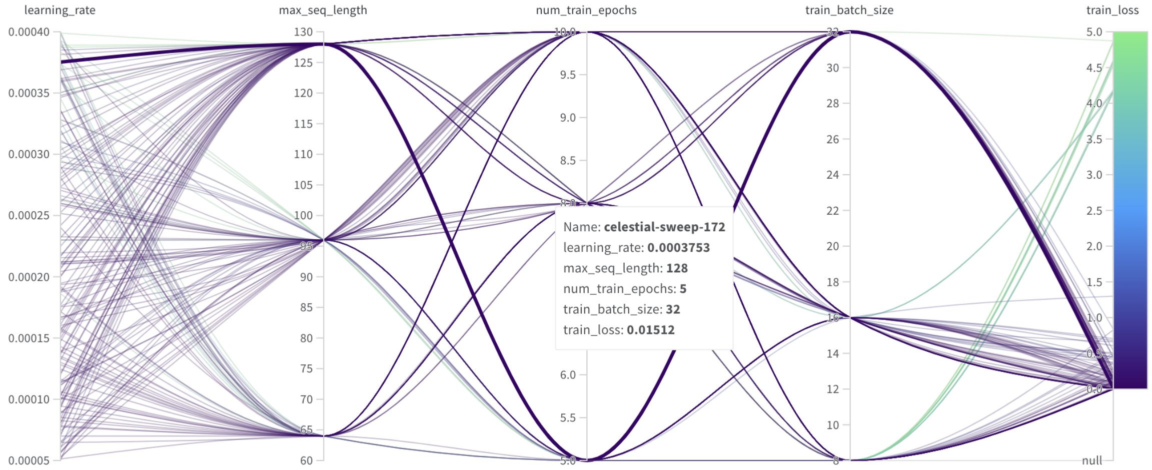
Method	pRR	exact match	f1
SEC	0.236	0.0262	0.112
S2S	-	0.306	-
FT-Arabert	0.250	0.0834	0.139

Table 2: Performance on Evaluation set

The best hyperparameters are shown below:

- Learning Rate: 3e-4
- Max Sequence Length: 128

[4] https://simpletransformers.ai/



Name: **celestial-sweep-172**
learning_rate: **0.0003753**
max_seq_length: **128**
num_train_epochs: **5**
train_batch_size: **32**
train_loss: **0.01512**

Figure 2: Hyper-Parameters(The Selected HP is the one used for submission.

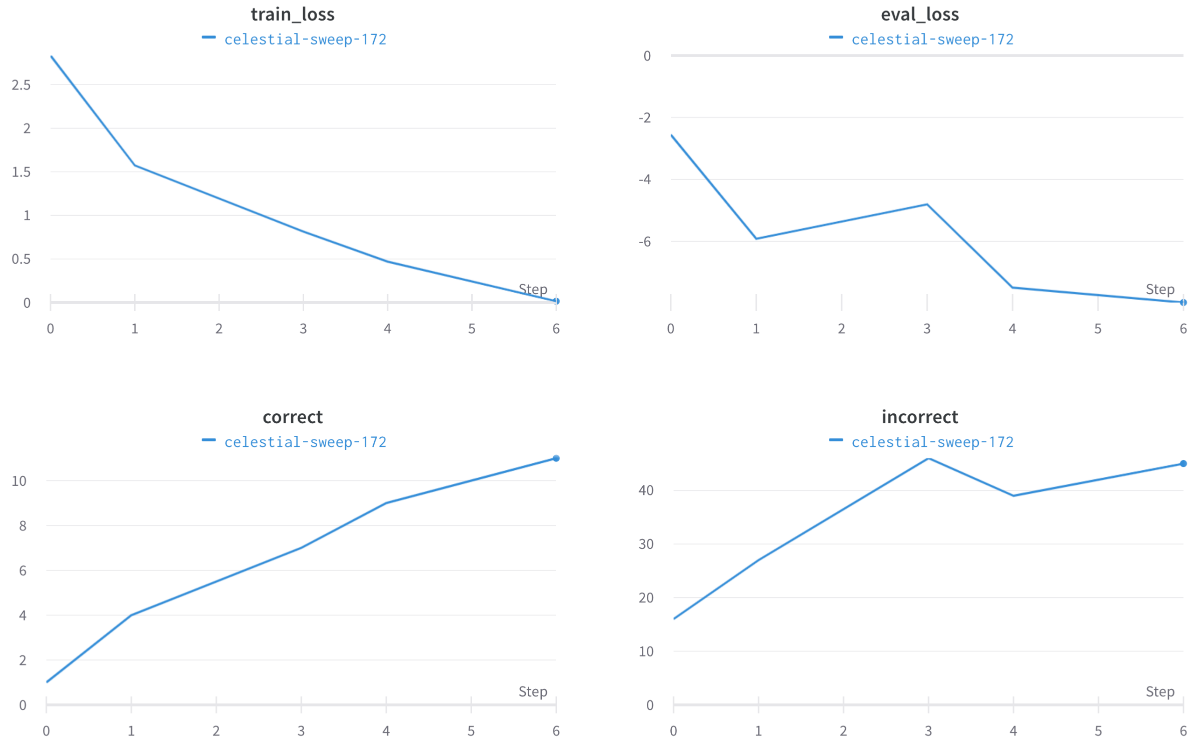


Figure 3: Results from the best performing Model on evaluation data.

- Number of Epochs: 5
- Training batch Size: 32

We Use Weights & Biases Sweeps for producing the visualizations. [5]

4. Results

The official evaluation measure of this shared task is Partial Reciprocal Rank. It is similar to Reciprocal Rank evaluation metric only that it considers partial matching with any of the gold labels. Two more metrics, Exact Match (EM) and F1@1 were also measured for checking if the top predicted answer matches one of the gold answers and the token overlap between the predicted and gold labels. Our best performing model has a pRR of 0.1913, Exact match of 4.2% and an F1@1 score of 9.1% .

[5] https://docs.wandb.ai/guides/sweeps

5. Error Analysis

After examining the predictions from the submitted model, we saw that our systems struggled significantly in answering the non-factoid questions whereas it did decently on factoid questions. The models for generating the contextual embeddings were traditional models used for resource-rich languages like English or German which receive an ample amount of domain-specific fine-tuning to do the closed domain retrieval task. The main reason we identify for the performance is the limited training data for a deep model like AraBERT. The model doesn't know the concept of question answering and has the weights adjusted as per the Self-Supervised MLM technique. Using just a 1000 samples to do the weight updations would have confused the model, leading to poor performance.

6. Conclusion and Future Work

Advancing the research on Under represented languages is very important and this Shared-Task seems to do just that. Arabic is a low resource language even though the number of people who use it is not low. We submit a simple method using a simple BERT based model with Arabic Corpus Pretraining. Due to the limited compute resources we just experimented with the mentioned hyper-parameters. However, we can optimize the number of layers of transformers in the model. Or, we can create custom parameter groups which can be used to set different learning rates for different layers in a model. Freeze layers or train only the final layer.

7. Bibliographical References

Abdelnasser, H., Ragab, M., Mohamed, R., Mohamed, A., Farouk, B., El-Makky, N., and Torki, M. (2014). Al-bayan: An Arabic question answering system for the holy quran. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 57–64, Doha, Qatar, October. Association for Computational Linguistics.

Alqahtani, M. and Atwell, E. (2016). Arabic quranic search tool based on ontology, June. © 2016, Springer International Publishing. This is an author produced version of a paper published in 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016 Proceedings, Natural Language Processing and Information Systems, Lecture Notes in Computer Science.. Uploaded in accordance with the publisher's self-archiving policy. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-41754-7_52.

Antoun, W., Baly, F., and Hajj, H. (2020). Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Fix, E. and Hodges, J. L. (1989). Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247.

Gusmita, R. H., Durachman, Y., Harun, S., Firmansyah, A. F., Sukmana, H. T., and Suhaimi, A. (2014). A rule-based question answering system on relevant documents of indonesian quran translation. *2014 International Conference on Cyber and IT Service Management (CITSM)*, pages 104–107.

Hakkoum, A. and Raghay, S. (2015). Advanced search in the qur'an using semantic modeling. *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–4.

Hakkoum, A. and Raghay, S. (2016). Semantic qa system on the qur'an. *Arabian Journal for Science and Engineering*, 41:5205–5214, 06.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28:129–136.

Malhas, R. and Elsayed, T. (2020). ¡i¿ayatec¡/i¿: Building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(6), oct.

Malhas, R., Mansour, W., and Elsayed, T. (2022). Qur'an QA 2022: Overview of the first shared task on question answering over the holy qur'an. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2020). mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.