Harnessing Multilingual Resources to Question Answering in Arabic

Khalid Alnajjar and Mika Hämäläinen
Rootroo Ltd
Helsinki, Finland
{firstname but don’t spam}@rootroo.com

Abstract

The goal of the paper is to predict answers to questions given a passage of Qur’an. The answers are always found in the passage, so the task of the model is to predict where an answer starts and where it ends. As the initial data set is rather small for training, we make use of multilingual BERT so that we can augment the training data by using data available for languages other than Arabic. Furthermore, we crawl a large Arabic corpus that is domain specific to religious discourse. Our approach consists of two steps, first we train a BERT model to predict a set of possible answers in a passage. Finally, we use another BERT based model to rank the candidate answers produced by the first BERT model.

Keywords: Question Answering Systems, Artificial Neural Model, Multilingual BERT, Arabic

1. Introduction

Question answering is natural language understanding problem that has received a fair share of attention in the past (Nishida et al., 2019; Rücklé et al., 2020; Asai and Choi, 2021). There are several datasets available for the task (Rajpurkar et al., 2018; Artetxe et al., 2020; Lewis et al., 2020). These datasets cover a very different domain than the one we are interested in this paper, namely the holy script Qur’an.

Qur’anic Arabic itself has also received its share of NLP interest (Sharaf and Atwell, 2012; Dukes and Habash, 2010; Alsaleh et al., 2021). There is even an earlier question answering system for Qur’an (Abdelnasser et al., 2014). Other historical Arabic texts have also received some research interest (Belinkov et al., 2016; Majadly and Sagi, 2021; Alnajjar et al., 2020).

In this paper, we describe our work on the Qur’an QA 2022 shared task data. The problem the QRCD (Qur’anic Reading Comprehension Dataset) dataset (Malhas and Elsayed, 2020) is built to solve is to predict an answer to a question given a passage in the Qur’an. The answer is within the passage, so the task for our model is to find where the answer starts and ends in the given passage.

The problem is a challenging one for several reasons. Firstly, Arabic language has greater degree of ambiguity in written form due to the fact that most of the diacritics are left out in writing. This means that several words that are pronounced different become homographs and have an identical written form. This ambiguity is not a characteristic of the language itself but much rather a result of the orthographic conventions. This ambiguity causes challenges not only in the dataset we are using but also in any pretrained Arabic language models.

Secondly, the publicly released part of the QRCD dataset is relatively small, consisting of only 710 samples in the training data and 109 samples in the development data. For this reason, we experiment with multilingual models and training data to alleviate this under-resourced scenario.

Thirdly, Arabic is a language with multiple dialects that are vastly different from each other. There is dedicated NLP research for several subdialects such as Tunisian (Ben Abdallah et al., 2020), Palestinian (Jarrar et al., 2014), Gulf (Adouane and Johansson, 2016) and Egyptian Arabic (Habash et al., 2012). These are very different from the Qur’anic Arabic we are focusing on, but they will be present in any large scale language model trained for Arabic on online corpora. For this reason, we needed to ensure that the model we use in this paper is trained exclusively on Modern Standard Arabic as it is the closest contemporary variant of the language to Qur’anic Arabic and there are no language models available for classical Arabic.

2. Dataset

The QRCD Qur’an question answering dataset consists of 1,337 question-passage-answer triplets, which is split into training (65%), development (10%), and test (25%) sets. As the amount of training data is small (i.e., 710 and 109 for training and validation, respectively), we leverage multilingual and crosslingual resources for question answering tasks while ensuring that the model is exposed to and aware of Islamic concepts.

To do so, we crawl multiple Islamic websites related to Tafseer (explanations of the Qur’an) and Fatwas (i.e., rulings or interpretations based on Islamic law for a given query) to build an Islamic-specific corpus. Table lists the web-sources we crawled and how many pages were retrieved per source. We crawled all the websites using Scrapy except for quran-tafseer.com for which we used Pytafseer. Additionally, we add

1 https://github.com/scrapy/scrapy
2 https://github.com/Quran-Tafseer/pytafseer
Table 1: The number of pages crawled per source in our Islam-specific corpus

| Source                  | N  |
|-------------------------|----|
| aliftaa.jo              | 907|
| binbaz.org.sa           | 23556|
| islamqa.info            | 14950|
| islamway.net            | 30978|
| islamb.net              | 30691|
| quran-tafseer.com       | 49888|

Table 2: The different question types, their frequency and statistics on the length of their corresponding answers (in tokens) in the training and development dataset

| Question | EN | N | min | avg | max |
|----------|----|---|-----|-----|-----|
| Where/How much | 56 | 1 | 10 | 55 |
| من أين؟ | من أين؟ | 255 | 1 | 6 | 58 |
| من أين؟ | من أين؟ | 107 | 1 | 8 | 225 |
| من أين؟ | من أين؟ | 1 | 12 | 12 |
| من أين؟ | من أين؟ | 104 | 1 | 10 | 32 |
| من أين؟ | من أين؟ | 174 | 1 | 12 | 62 |
| من أين؟ | من أين؟ | 2 | 1 | 2 | 3 |
| من أين؟ | من أين؟ | 14 | 3 | 11 | 27 |

Steps along with any preprocessing and postprocessing phases.

3.1. Domain adaptation

As the model we are basing our work on (i.e., multilingual BERT) is trained on a generic encyclopedia corpus (Wikipedia) and has little exposure to Islamic and Qur’anic concepts, we continue training the multilingual BERT model to adapt it to the domain of the task here. In our previous research (Hamalainen et al., 2021a; Hamalainen et al., 2021b), we have found that BERT based models tend to work better if their training data has had text of a similar domain as the downstream task the model is fine-tuned for. Therefore, we believe that domain adaptation is beneficial in this case as well.

We convert the crawled data into a textual corpus, which we clean from non-Arabic text and remove any Arabic diacritics or punctuation using UralicNLP (Hamalainen, 2019). In the case of Fatwas, we format the text as question first followed by the answer provided by the Mufti, and for Tafseer, we add the context (i.e., passage/verse) prior to the question. Qur’an data is added as it is. The textual corpus is then split into 80% training and 20% validation. We train the base model for the task of Masked Language Modeling and Next Sentence Prediction for 3 full epochs on our entire textual corpus.

3.2. Fine-tuning for Question Answering

Here, we fine-tune our BERT model for the task of Qur’an question answering. The data used for fine-tuning comes both from the QRCDD dataset itself and the additional QA datasets (MLQA, SQuAD and XQuAD). The purpose of the other QA datasets is to make the model learn better the task of question-answering and make it learn to use multilingual information better for this task. In essence, this should improve the results for Arabic even though a majority of the training data was in a different language.

We append a fully-connected dense layer that accepts the BERT’s hidden layers and outputs two vectors of predictions. These vectors are predictions for the start and end positions of the predicted answer in the context. We use the Adam algo-

3https://github.com/aliftype/quran-data
### 3.3. Predicting Answers and Post-processing them

When inferring answers, we clamp the results to the number of tokens in the context and we ignore any special tokens prior to applying the softmax function on the predicted positions.

We apply a post-processing step on the top $N$ generated answers. This goal of this step is to 1) ensure that the length of the answer corresponds to the type of question that is being asked and 2) eliminate overlapping predictions.

Different types of questions require answers of different lengths; for instance, answers to “who”-questions will most of the time be shorter (typically a single word is sufficient to answer it) than answers to “why”-questions (requires further elaboration consisting of several tokens). For this reason, we apply some data analysis on the Quran QA dataset to find out what the questions are that are present in the dataset and what the minimum, average and maximum lengths of the answers are. We use Farasa segmenter \cite{Abdelali2016} to process the data for this analysis.

As a given type of questions might be expressed in different ways, we have applied some manual clustering to group all asked questions in the dataset into 8 types based on the interrogative pronouns used. Table\ref{table:questions} presents the question types, how many times they were present in the dataset and statistics on their answers. All predicted answers that are smaller than the average answer length are extended to either the nearest full-stop that marks the end of the verse or the average length, whichever is shorter.

When predicting multiple answers for a question, it is commonly the case that the model would predict overlapping answers. In such cases, we merge them together by taking the smaller starting position and maximum end position.

### 3.4. Similarity Recommendation

We noticed during our observations of the Qur’an QA dataset that some questions are detailed and elaborate further on what is being sought as an answer. Such elaborations would indicate that the most semantically similar verse to the question is probably the answer to it. For this reason, we use the second version of AraBERT Large \cite{Antoun2020} to extract the features of the question and all verses in the given passage. Thereafter, we find the most similar verse to the question by applying the cosine similarity on the extracted features. The most similar verse is appended to the list of answers if it was not predicted already by the model and there are less than 5 predicted answers.

### 4. Results and Evaluation

We experiment with different models and techniques for predicting answers. First, we test out different BERT models and compare them to our custom model. Secondly, we investigate the effects of fine-tuning our model with different training and validation question and answering datasets along with the Quran QA dataset. Lastly, we assess the benefits of post-processing the predictions and including the most similar verse to the question as an answer. In our tests, we evaluate the models based on the metrics that are considered in the shared task, namely partial Reciprocal Rank (pRR) \cite{Malhas2020}, exact match and F1 \cite{Rajpurkar2016}.

### Table 3: Experimental settings and their performances

| # | Model | Training Data | pRR | Exact Match | F1@1 |
|---|-------|---------------|-----|-------------|------|
| 1 | Base  | Quran QA training | 0.347 | 0.009 | 0.311 |
| 2 | Base  | Quran QA training + SQuAD + MLQA + XQuAD | 0.351 | 0.092 | 0.288 |
| 3 | Model 2 | Quran QA training | 0.373 | 0.028 | 0.345 |
| 4 | Model 2 | Quran QA training + Quran QA development | 0.540 | 0.092 | 0.526 |
| 5 | Model 4 + post-processing | 0.700 | 0.358 | 0.688 |
| 6 | Model 4 + post-processing + recommendation | 0.704 | 0.358 | 0.688 |
| 7 | Model 3.2 + Model 4 + post-processing + recommendation | 0.648 | 0.211 | 0.639 |

### Table 4: Different BERT models trained and evaluated solely on the Quran QA dataset

| Models | pRR | Exact Match | F1@1 |
|-------|-----|-------------|------|
| KUISAIL’s base BERT \cite{Safaya2020} | 0.286 | 0.037 | 0.236 |
| KUISAIL’s large BERT \cite{Safaya2020} | 0.330 | 0.037 | 0.278 |
| CAMEL-BERT Quarter \cite{Khor et al. 2021} | 0.215 | 0.018 | 0.222 |
| Our BERT model | 0.347 | 0.009 | 0.311 |

### Table 5: Results on the test set

| Run | pRR | Exact match | F1@1 |
|-----|-----|-------------|------|
| Run 6 | 0.392 | 0.113 | 0.354 |
| Run 7 | 0.409 | 0.092 | 0.364 |
Comparing KUISAIL’s base and large models suggest that bigger models improve the performance of the model. However, larger models require a longer time to train, and for this reason we opted for using a base model. Despite using a smaller multilingual model as a base model, adapting it to the domain of this task has clearly improved the quality of its predictions. All results presented after this point use our custom BERT model.

In Table 5 we can see the results of our two systems when comparing to the official test set of the shared task. When we compared our system on a question level to the median values across all the submissions to the task, we found that over half of the time our best system achieved better scores than the median value. However, our best model had the highest possible score among all submissions around 15% of the time. Interestingly, our worst model had the highest possible score among all the submissions 17.6% of the time despite having poorer overall performance.

Table 6 lists the different settings we experimented with and their evaluation results on the development split. All the models had the Quran QA dataset as the validation dataset during the training phase, and they have been trained for 3 epochs. By comparing the first and second settings, we see that including other question and answering datasets during the training phase improves the predictions. Fine-tuning the model that has been exposed to other question and answering datasets further using Quran QA dataset only outperforms using Quran QA dataset solely, which demonstrates the great importance of utilizing relevant linguistic resources in other languages and applying domain adaptation. In the 4th experimental setting we include the Quran QA development split in the training dataset to cover as many cases as possible given that the amount of training data is very small; despite it being a non-recommended practice.

Our experiments point out that post-processing the predicted answers to ensure that they are of an adequate length based on the question type and that no overlapping answers have been predicted boosts the results from pRR of 0.54 to 0.7. Including the most similar verse to the question as a possible answer raises the results by a bit but it does not affect them negatively.

From our observations of answers predicted by model #6 and #7 is that sometimes we would predict different answers where one of them is correct. To benefit from both of the models and include their variations in the answers, we consider answers produced by them and remove any overlapping answers during the post-processing phase. We have submitted two runs to the shared task, which are experiments number 6 and 7. Looking at examples of generated answers by our mod-
els in Table 5 illustrates cases where the predictions have been fully accurate by predicting an exact match (e.g., example #2) and partially correct (e.g., examples #1, #3 and #4). For the partially correct predictions, the model either predicts lengthier or shorter answers which could be due to the post-processing phase. However, we find that the length of gold answers is subjective. The fifth case is an example of wrong predictions; nonetheless, the top prediction is includes two nouns and the question is asking for people (whom) which tells us that the model made its best guess given the context and it was very close.

5. Conclusions

In conclusion, we have embraced multilingual models, and question and answering resources to build a question answering model for Qur’an. Our results indicate that applying domain adaptation and fine-tuning the model with relevant data sets increases the performance of the models, especially in the case of limited training data like this one.

As the models predict the start and end positions of the answer in the context, it is very likely that the predictions are off by few tokens. Post-processing the predictions to correspond to the expected answer per question type and merging any overlapping cases had a huge boost on the quality of predictions.

6. Bibliographical References

Abadji, J., Ortiz Suarez, P., Romary, L., and Sagot, B. (2022). Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. arXiv e-prints, page arXiv:2201.06642, January.

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. In 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 11–16. Association for Computational Linguistics.

Abdelnasser, H., Ragab, M., Mohamed, R., Mohamed, A., Farouk, B., El-Makky, N., and Torki, M. (2014). Al-hayan: 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.

Adouane, W. and Johansson, R. (2016). Gulf Arabic linguistic resource building for sentiment analysis. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16), pages 2710–2715, Portoroz, Slovenia, May. European Language Resources Association (ELRA).

Alnajjar, K., Hämäläinen, M., Partanen, N., and Rueter, J. (2020). Automated prediction of medieval arabic diacritics. arXiv preprint arXiv:2010.05269.

Alsaeleh, A., Atwell, E., and Altabhhan, A. (2021). Quranic verses semantic relatedness using AraBERT. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 185–190, Kyiv, Ukraine (Virtual), April. Association for Computational Linguistics.

Antoun, W., Baly, F., and Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pages 9–15, Marseille, France, May. European Language Resource Association.

Asai, A. and Choi, E. (2021). Challenges in information-seeking QA: Unanswerable questions and paragraph retrieval. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1492–1504, Online, August. Association for Computational Linguistics.

Belinkov, Y., Magidow, A., Romanov, M., Shnidman, A., and Koppel, M. (2016). Shamela: A large-scale historical Arabic corpus. In Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH), pages 45–53, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Ben Abdallah, N., Kchaou, S., and Bougaress, F. (2020). Text and speech-based Tunisian Arabic sub-dialects identification. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 6405–6411, Marseille, France, May. European Language Resources Association.

Dukes, K. and Habash, N. (2010). Morphological annotation of quranic Arabic. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10), Valletta, Malta, May. European Language Resources Association (ELRA).

Habash, N., Eskander, R., and Hawwari, A. (2012). A morphological analyzer for Egyptian Arabic. In Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology, pages 1–9, Montréal, Canada, June. Association for Computational Linguistics.

Hämäläinen, M., Alnajjar, K., Partanen, N., and Rueter, J. (2021a). Never guess what I heard... rumor detection in Finnish news: a dataset and a baseline. In Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, pages 39–44, Online, June. Association for Computational Linguistics.

Hämäläinen, M., Patpong, P., Alnajjar, K., Partanen, N., and Rueter, J. (2021b). Detecting depression in thai blog posts: a dataset and a baseline. In Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021), pages 20–25.
Hämäläinen, M. (2019). Uralicnlp: An nlp library for uralic languages. *Journal of open source software*.

Inoue, G., Alhafni, B., Baimukan, N., Bouamor, H., and Habash, N. (2021). The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online), April. Association for Computational Linguistics.

Jarrar, M., Habash, N., Akra, D., and Zalmout, N. (2014). Building a corpus for palestinian Arabic: a preliminary study. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27, Doha, Qatar, October. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. In *arXiv*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. In *arXiv*.

Majadly, M. and Sagi, T. (2021). Dynamic ensembles in named entity recognition for historical Arabic texts. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 115–125, Kyiv, Ukraine (Virtual), April. Association for Computational Linguistics.

Mäkelä, E., Lagus, K., Lahtli, L., Säily, T., Tolonen, M., Hämäläinen, M., Kaislanieme, S., Nevalainen, T., et al. (2020). Wrangling with non-standard data. In *Proceedings of the Digital Humanities in the Nordic Countries 5th Conference Riga, Latvia, October 21-23, 2020. CEUR-WS.org*.

Malhas, R. and Elsayed, T. (2020). *Ayatec*: 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.

Nishida, K., Nishida, K., Nagata, M., Otsuka, A., Saito, I., Asano, H., and Tomita, J. (2019). Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy, July. Association for Computational Linguistics.

Rücklé, A., Pfeiffer, J., and Gurevych, I. (2020). MultiCQA: Zero-shot transfer of self-supervised text matching models on a massive scale. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2471–2486, Online, November. Association for Computational Linguistics.

Safaya, A., Abdullatif, M., and Yuret, D. (2020). KUI-SAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online), December. International Committee for Computational Linguistics.

Sharaf, A.-B. and Atwell, E. (2012). QurAna: Corpus of the quran annotated with pronominal anaphora. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 130–137, Istanbul, Turkey, May. European Language Resources Association (ELRA).

7. Language Resource References

Arteste, Mikel and Ruder, Sebastian and Yogatama, Dani. (2020). *On the Cross-lingual Transferability of Monolingual Representations*. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).

Lewis, Patrick and Oguz, Barlas and Rinott, Ruty and Riedel, Sebastian and Schwenk, Holger. (2020). *MLQA: Evaluating Cross-lingual Extractive Question Answering*. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Malhas, Rana and Elsayed, Tamer. (2020). *Ayatec*: building a reusable verse-based test collection for arabic question answering on the holy qur’an. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP).

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November. Association for Computational Linguistics.

Rajpurkar, Pranav and Jia, Robin and Liang, Percy. (2018). *Know What You Don’t Know: Unanswerable Questions for SQuAD*. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).