Proceedings of the OSACT 2022 Workshop @LREC2022, pages 96–103
Marseille, 20 June 2022

eRock at Qur'an QA 2022: Contemporary Deep Neural Networks for Qur'an based Reading Comprehension Question Answers

Esha Aftab, Muhammad Kamran Malik
Department of Computer Science, Department of Information Technology
Faculty of Computing and Information Technology, University of the Punjab, Lahore, Pakistan
esha.aftab@pucit.edu.pk, kamran.malik@pucit.edu.pk

Abstract

Question Answering (QA) has enticed the interest of NLP community in recent years. NLP enthusiasts are engineering new Models and fine-tuning the existing ones that can give out answers for the posed questions. The deep neural network models are found to perform exceptionally on QA tasks, but these models are also data intensive. For instance, BERT has outperformed many of its contemporary contenders on SQuAD dataset. In this work, we attempt at solving the closed domain reading comprehension Question Answering task on QRCD (Qur'anic Reading Comprehension Dataset) to extract an answer span from the provided passage, using BERT as a baseline model. We improved the model's output by applying regularization techniques like weight-decay and data augmentation. Using different strategies we had 59% and 31% partial Reciprocal Ranking (pRR) on development and testing data splits respectively.

1. Introduction

Question Answering (QA) is a longstanding task in Natural Language Processing that aims at providing a reliable, precise and well-formed natural language answer to a meaningful question from a given text body. The resurgence of Question Answering as an appealing problem in recent years, can be mainly attributed to the following factors; a) efficient Information Retrieval (IR) systems; b) neural Reading Comprehension (RC) models; c) availability of large scale annotated datasets. Another aspect that makes the QA task worth the effort is its wide range of potential applications. Some of the thriving and far reaching utilities are in personal assistant apps and chat bots to respond frequently asked questions in real time. Google Assistant [1] and FAQ Bot [2] are a few examples of such systems.

With ever increasing online resources and a large number of users looking for reliable and exact answers to their queries, it is a pressing need of time for modern search engines to employ intelligent automated QA models. It will expedite the process of examining the massive amount of data from web or local repositories and fetch rational and relevant responses to queries with higher degree of precision.

In classic search engines, the information retrieval systems extract keywords from a query and look them up by crawling through web documents to find similar resources. The algorithms like TF-IDF and BM25 are used to determine keyword frequency in crawled web documents and rank them with respect to their relevance to the query, respectively. The resulting links are displayed in the order of their computed relevancy. Many popular search engines like Google (Falconer, 2011) and Bing have shifted their paradigm towards providing precise answers to the queries rather than just links.

The Question Answering systems can be divided into two major categories: a) Open-domain Question Answering (OpenQA) – where the questions may belong to any topic or genre from a knowledge base e.g. Wikipedia; b) Closed-domain Question Answering (ClosedQA)– where the questions belong to a specific knowledge field or a particular genre e.g. law, education, finance, weather etc. The OpenQA tasks are more prevalent in studies, than closedQA, owing to their wider topic range and numerous resources.

The operational pipeline of either type of QA system, may undergo following sequential steps a) Query analysis b) Search and information retrieval and c) Answer formulation. The pipeline subtask 'Answer Formulation' further divides the Question Answering systems into two categories: a) Abstractive Question Answering – with well-formed natural language answers abstracted from the relevant text without copying exact phrases from it, b) Extractive Question Answering – with answers derived as a consecutive sequence of tokens from the relevant documents. Abstractive Question Answering is more suitable when complex reasoning is required over multiple paragraphs/documents to answer a question (Chen et al., 2019).

Machine Reading Comprehension (MRC) is a sub-task of OpenQA (Ruder, 2021). According to (Chen, 2018) MRC is a task of understanding unstructured text and precisely answer any questions about it. The answer is extracted from the context document by highlighting the start and end of the span. In this work, we attempt at solving the a closed domain MRC task released in 5th workshop on Open-Source Arabic Corpora and Corpora Processing Tools (OSACT) in LREC 2022 (Malhas et al., 2022). An annotated Qur'anic Reading Comprehension Dataset (QRCD) is shared at the gitlab repository[3] containing question-answer-passage tuples. The task objective is to extract an answer span against a question, from the provided passage of Quran. It is required to generate 5 possible answer spans ranked according to the scores assigned by our specific

[1]https://assistant.google.com/

[2]https://www.theta.co.nz/technologies/faq-bot/

[3]Rana Malhas and Tamer Elsayed. Official Repository of Qur'an QA Shared Task. https://gitlab.com/bigirqu/quranqa. February 2022

model. The performance of the model is evaluated over three metrics; Exact Match (EM), F1 score and partial Reciprocal Ranking (pRR). We contribute to effectively solve the task by adopting following approaches:

- Using a BERT (Devlin et al., 2018) pre-trained language model and fine tuning it for the specific dataset QRCD.
- Using regularization technique, Decoupled Weight Decay (Loshchilov and Hutter, 2017), to avoid overfitting.
- Using two different pre-trained models: a) BERT pre-trained Arabic model, b) BERT pre-trained multilingual model; showing that the first pre-trained model has an edge over the second one and out performs it.
- Using data augmentation to increase training data. For this purpose, we used Annotated Corpus of Arabic Al-Quran Question and Answer (AQQAC) (Alqahtani and Atwell, 2018a), filtered it to select the question answer tuples that matched in style to that of QRCD. For each tuple, we took the sequence of Quran verses present in answer and generated their context passage from Quran's simple text downloaded from Tanzil project [4].
- Finally, we analyze that how much a model can hatch improved results using regularization techniques like weight decay and data-augmentation. (Note: the augmented data being collected with an altogether different methodology might not be true representative of the testing data).

2. Related Work

Neural Machine Reading Comprehension (MRC) has gained success over answer retrieval from unstructured text. Many statistical systems like AskMSR (Banko et al., 2002) exploit the answer redundancy in the source data. In MRC, the answer might not be redundant or might occur just once, thus leading to undesired results. Thus a model needs to acquire deeper understanding of the question, the context passage and their mutual relationship. The neural models address this need by learning question and context similarities. One such model is BiDAF (Seo et al., 2016) based on Recurrent Neural Network (RNN) and uses bidirectional attention flow to generate query-aware-context representations. It took lead on SQuAD (Rajpurkar et al., 2016) dataset at the time of submission, with Exact Match (EM) and F1 scores of 68% and 77% respectively. DrQA (2017) (Chen et al., 2017) is a multi-layer RNN model. It presents pipeline architecture with document retriever, employing TF-IDF algorithm, and the document reader modules. The DrQA reader module showed EM score of 69% and F1 score of 78.8% on SQuAD. QANet (Yu et al., 2018) uses convolution and self-attention mechanisms making it more efficient than counterpart RNN based models. Being faster the model trains on more data generated through back translation.

Generative Pre-trained Transformer (GPT-1) (Radford et al., 2018) model, with 117M parameters, introduced the approach of pre-training on unlabeled data with unsupervised tasks and fine-tuned on downstream tasks with remarkable results regardless of the limited size of task specific training data. Many more powerful models followed the pursuit. A few to name are BERT (Devlin et al., 2018), XLNET (Yang et al., 2019) and T5 (Raffel et al., 2019). GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), the successors of GPT-1, were trained on even larger data and with significantly more parameters, 1.5 billion and 175 billion parameters respectively.

BERT (Devlin et al., 2018) has multilayer bidirectional transformer encoder architecture that pre-trains on two unsupervised tasks Masked Language Model (MLM) or Next Sequence Prediction(NSP) model. RoBERTa (Liu et al., 2019) an extension of BERT is trained for longer period over longer sequences, with NSP loss removed under the observation that it is only useful if input sequences are individual sentences instead of passages with multiple sentences. A significant improvement is observed in the performance on SQUAD 1.1/2.0, MNLI-m, SST-2 and RACE datasets in comparison to original BERT and even exceeds many of the successors of BERT.

Dataset	Source	Size
DAWQUAS (Ismail and Homsi, 2018)	News, social, women, science and technology websites	3205
AQQAC (Alqahtani and Atwell, 2018a)	Book: 1000 Su'al Wa Jawab Fi ALKORAN, Website based on Al-Quran and Tafseer	2224 public: 1224
Arabic SQuAD (Mozannar et al., 2019)	Wikipedia translation of original SQuAD1.1	48,344
ARCD (Mozannar et al., 2019)	Arabic Wikipedia	1,395
TyDiQA-GoldP (Clark et al., 2020)	Wikipedia	204K
Ayatec (Malhas and Elsayed, 2020)	The Holy Quran verified text from Tanzil Project	1,762
AQAD (Atef et al., 2020)	Arabic Wikipedia articles matching SQuAD1.1 articles	17,911

Table 1: Arabic Question Answering Datasets

As mentioned earlier in section 1 that one of the driving forces in popularity of QA task is availability of the bench-

[4] https://tanzil.net/download/

mark datasets. . SQuAD (Rajpurkar et al., 2016) is the most popular MRC dataset in English with 107,785 question-answer pairs over 23,125 paragraphs extracted from 536 Wikipedia articles selected after multiple ranking and random sampling. SQuAD2.0 (Rajpurkar et al., 2018) contains 53,775 unanswerable questions but seemingly answerable authored by crowd workers. English being a resourceful language has numerous evaluation datasets publically available both for MRC and OpenQA e.g. HotpotQA (Yang et al., 2018), Natural Questions (NQ) (Kwiatkowski et al., 2019), triviaQA (Joshi et al., 2017) etc. However QA research in resource poor languages like Arabic, Persian and Urdu is severely hampered. One ingenious resort to overcome this deficiency is through machine translation now much more advance and accurate owing to neural machine translation (NMT) models. For instance some of the datasets constructed using this method are PQuAD (Darvishi et al., 2022), K-QuAD (Lee et al., 2018) and SQuAD-es (Carrino et al., 2019) that are the respective translations of SQuAD dataset in Persian, Korean and Spanish languages.

(Mozannar et al., 2019) presents the Arabic translation of a subset of SQuAD containing 48,344 questions on 10,364 paragraphs. In addition to Arabic-SQuAD, the study also contributes another smaller Arabic Reading Comprehension Dataset (ARCD) containing 1,395 questions compiled over Arabic Wikipedia articles through crowdsourcing. It presents system for open domain question answering in Arabic (SOQAL) having Retriever and Reader modules. On both Arabic-SQuAD and ARCD, the reader module employs BERT-Base un-normalized multilingual model and QANet fastText, with the former taking the lead. Independent testing on ARCD and Arabic-SQuAD using BERT shows very insignificant difference in results, increasing confidence in data generated through NMT. DAWQUAS (Ismail and Homsi, 2018) is a collection of 3205 why-question-answers pair collected using Google Search API to look up for web pages in general as well as some specific news and social sites that contained the Arabic word ***'limadha'*** (meaning: why). TyDiQA-GoldP (Clark et al., 2020) is multi-lingual collection of 204K question-answer pairs for 11 languages. The data source is Wikipedia and questions are authored by human annotators with little access to article's content. Answers are generated later by annotators with full access to the content of article by selecting best answer passage and a minimal span in that passage if possible. Arabic Question-Answer dataset contains 17,911 questions from 3,381 paragraphs out of 299 Arabic Wikipedia articles. Only those articles and paragraphs are selected from Arabic Wikipedia that are also present in English-to-Arabic translation of SQuAD2.0 articles and paragraphs. The questions are generated for selected paragraphs through machine translation of SQuAD2.0 questions.

Annotated Corpus of Arabic Al-Quran Question and Answer (AQQAC) (Alqahtani and Atwell, 2018a) is a collection of 2224 question-answers and other helpful details about the Holy Quran from two authentic sources, the book "1000 Su'al Wa Jawab Fi ALKORAN" and a website on information about Quran.

AyaTEC is a test collection of verse based question-answers from the Holy Quran comprising of 207 question over 11 categories of the Holy Quran. The questions are gathered from Arabic QA systems on Quran and the users. Two user categories are defined: Curious – asking questions related to teachings of the Quran, Skeptical – asking controversial questions. The answer space is restricted to Qura'nic verses. The dataset provides all verses of Quran, exhaustively that may answer a question, collected by two freelancers with the knowledge of Quran. The relevance of selected verses of Quran to the question was verified by three specialists in the Holy Quran.

All the Arabic datasets discussed in this section are summarized in Table-1

3. Approach for Quran QA Task:

We use BERT (Devlin et al., 2018) for QuranQA task as our baseline model. BERT input can represent both a single sequence of text and a pair of two sequences separated by a special token '[SEP]' depending on the nature of downstream task. For instance, in our case we pass on the question-passage pair for our QA task.

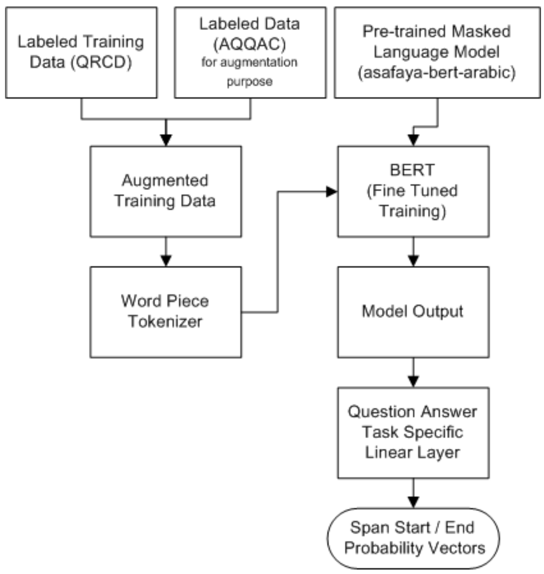

Figure 1: Our BERT QA Pilpeline Approach

Our model has a pipeline architecture, as shown in Figure-1, starting off with data preparation. The QRCD training data needs to be processed to transform into a format that can be fed as input to BERT. The given dataset has 118 unique questions, accompanied by multiple context passages from Quran and multiple possible answers from each passage. The answer text and its starting position in the passage is listed in the dataset. The ending position of an answer in the context needs to be known too for model to yield a span in terms of probabilities of start and end indices of the passage. Therefore the data is processed to create unique question-answer-passage tuples in a format consistent with BERT input.

After the pre-processing step QRCD training data resolves into 861 unique question-answer-passage tuples. Due to

the limited size of training data and BERT being a hefty model, it tends to over fit very quickly. In order to keep this problem at bay, we augmented the training data with 473 additional tuples acquired from AQQAC dataset after careful sampling and building context passages from Quran which were deficit in the original dataset.
Entailing step was to initialize BERT either with some initial configurations or with some generic pre-trained checkpoint. For pre-training, BERT opt for two unsupervised learning schemes: Masked Language Modeling (MLM) and Next Sequence Prediction (NSP). We initialized BERT with two different pre-trained MLM models, 1) BERT multilingual base model[5], 2) asafaya/bert-base-arabic[6]. We empirically show that mono-lingual model is best fitting in our case.
In the subsequent step, question-passage pairs in the training data undergo tokenization routine. The Word Piece tokenizer is employed to generate the word embeddings with a special opening token '[CLS]', and another special token ['SEP'] parting the question and context tokens as well as indicating the end of input sequence. The model feeds off the word embeddings to fine-tune and learns query to context segmentation and positional embeddings. The model outputs are two vocabulary sized vectors representing probabilities of each token as a potential start and end positions of the answer span in the context.

4. Experimental Setup

This section describes the datasets used, model's output and evaluation measures used for experiments.

4.1. Dataset

We evaluate our models on the QRCD (Qur'anic Reading Comprehension Dataset) dataset (Malhas and Elsayed, 2020) shared at gitlab repository of Qur'an QA Task. The dataset includes 1093 question-passage pairs and along with their exhaustively extracted answers which results in 1,337 question-passage-answer triplets. The dataset is split into training (65%) , validation/development (10%) and test (25%) sets. The dataset is shared in JSON file with each line having passage, question, answer/answer-list, chapter number and list of verse numbers from passage.

4.2. Model Settings

The BERT pre-trained models **bert-base-multilingual-uncased**, supporting 102 languages, and **asafaya/bert-base-arabic** are both trained with 12 hidden layers, 768 hidden size, 12 attention heads, 0.1 dropout on each hidden layer and 110 parameters. For fine-tuning the Model, we run 30 epochs to train, using a batch size of 8.

4.3. Model Output

The QuranQA task requirement is to provide 5 probable answer spans ranked 1 (highest) to 5 (lowest) according to their scores. Each answer is listed in order of its rank providing following information: anwer text, rank and the score computed by the model.

4.4. Evaluation Method

For evaluation purpose, three metrics are used.

- Exact Match metric is applied only to the highest ranking answer. It awards a binary score 1 or 0. The score is 1 on finding output answer text in the context passage, 0 otherwise. Arabic prefixes and punctuations are removed from the answer and the passage, before finding exact match.

- The F1 metric, again applied to the highest ranking answer only, measures the average word overlap between the predicted and the ground truth answer.

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

$$precision = \frac{overlapping\_token\_count}{token\_count\_in\_predicted\_answer} \quad (2)$$

$$recall = \frac{overlapping\_token\_count}{token\_count\_in\_ground\_truth} \quad (3)$$

- pRR is the official metric of Qur'an QA Task. If an answer is assigned zero F1 score against any of the ground-truth answers then the average F1 scores of answers next in rank are taken into account. While coursing through answers with lower ranks, the F1 score of each answer is penalized by taking its product with the reciprocal of answer's rank.

5. Experiments

This section emphasizes on our experiments for QuranQA task. We begin with a baseline BERT model to extract preliminary results and improve results with subsequent modifications to the baseline model. We employ fine-tuning, a transfer learning strategy, using two differently pre-trained models. Finally we apply regularization strategies, data augmentation and weight-decay to achieve enhanced results.

5.1. Baseline Model

A BERT model is set up from scratch, with initial default settings same as mentioned for pre-trained models in section 4.2. The model is supplied with training data token embeddings, generated as discussed in section 3, as input.

5.2. Pre-trained Models and Fine-Tuning

Taking into account the size of training dataset, initializing BERT from pre-trained checkpoints give a reasonably enhanced kick start. The model can fine-tune with task specific training data and reaps astounding results. We selected two pre-trained BERT models for this task. First is the **bert-base-multilingual-uncased** pre-trained for 102 different languages comprising a vocabulary of 105879 tokens, open sourced at Hugging Face online library. The second pre-trained model is **asafaya/bert-base-arabic** (Safaya et al., 2020), for Arabic only with 32000 tokens, downloaded from Hugging Face.

[5]https://huggingface.co/bert-base-multilingual-uncased
[6]https://huggingface.co/asafaya/bert-base-arabic

5.3. Data Augmentation

The magnitude of our training data has its limitations to fully exploit the strength of a deep neural network model like BERT and is prone to over-fit. We regularize the model by supplementing it with additional training data. This additional data may not be the true representative of the development and test data splits of QRCD; nevertheless, it supports more generalized learning of the model parameters and keep it from over-fitting. We've used AQQAC (Alqahtani and Atwell, 2018a) originally containing 2224 annotated questions-answer pairs and only 1224 released publically due to copyright concerns. Each question-answer pair also provides ancillary information like question ID, question opening word, relevant chapter and verse numbers, question topic, question type, Al-Quran ontology concepts (Alqahtani and Atwell, 2018b) and question source.

The two datasets, QRCD and AQQAC, are collected using different methodologies and annotated for different objectives. Therefore we filter AQQAC to extract only those data points that best match in style to QRCD. We also construct the context passages as they are not already available in the original dataset.

For question-answer sampling from AQQAC we observed following salient features of the data. Unlike QRCD, the answer to a question is not necessarily a Quranic verse or a part of it. Many of the answers to questions are in plain language using Modern Standard Arabic (MSA) and supplement their argument with Quranic verses. Another very common type of questions are those seeking an explanation or interpretation to Quranic verses. The corresponding answers simply explain the verses but does not explicitly contain or refer to other relevant verses. Pertaining to these dissimilarities some preprocessing on AQQAC is inevitable.

Our objective is to identify the pairs where the answer is a part of a verse, a complete verse, or a continuous sequence of multiple verses. We use following steps to achieve this.

- On examining the data property **'question type'** we found that the two categories in this field ***'ashrah'*** (meaning: explain) and ***'fasr'*** (meaning: interpret) are mostly an explanation or interpretation of the given Quranic text, in modern standard Arabic. Therefore we omit these questions.

- In the remaining data, we use an automated routine to search the questions with answers containing no verse from the Holy Quran or any reference to one. We omit such question-answers too as our minimal criteria of selection is that the answer should contain at least one verse from the Holy Quran.

- After the first two steps, remaining questions have at least one or multiple continuous verses either as a direct answer or as a reference to support the answer. In either case, any plain text appended before and after the verses is removed. In the severed text, the verse numbers are replaced by sentence delimiters.

We get a set of 473 question-answers out of 1224. To make the selected set usable in QuranQA task we need a context passage along with question answer pairs. We use this idea that each answer is either a single verse or a set of continuous verses from Quran so their context can be taken from Quran too. We used simple text of Quran from the Tanzil project for this purpose. To build the passage for each data point, we take verses from the answer and locate them in the text of Quran. Then we select two verses from the preceding and following contexts each and concatenate them before and after the verses taken from the answer respectively. This gives a reasonable context for our task.

We also provide the index of passage after the preceding context, from where the original answer begins as a starting position of the answer span. After curating this information for each data point we save it in the same format as the QRCD training data, ready to use.

5.4. Weight Decay Regularization

Apart from data augmentation, to train model more generically we attempted at using the weight-decay (Loshchilov and Hutter, 2017) with Adam optimizer. The range of values we tested it for are 0.1, 0.01 and 0.001 with 0.01 giving the best results of the three.

6. Results and Discussions

In this section we shall discuss the results of our approach and analyze the impact of settings and techniques we applied to make it better. We refer to each model setting with a short code name as follows:

R0: Baseline model trained from scratch
R1: Fine-tuning (*bert-base-arabic*)
R2(a): Fine-tuning (*bert-base-arabic*) + Data augmentation
R2(b): Fine-tuning (*bert-base-multilingual-uncased*) + Data augmentation
R3: Fine-tuning (*bert-base-arabic*) + Weight decay (0.01)
R4(a): Fine-tuning (*bert-base-arabic*)+ Weight decay (0.01) + Data augmentation + Shuffled training data
R4(b): Fine-tuning (*bert-base-arabic*)+ Weight decay (0.01) + Data augmentation
R5: Fine-tuning (*bert-base-arabic*)+ Weight decay regularization (at value 0.01) + Data augmentation + Training data inclusive of Development data for training

Table 2, Table 3 and Table 4 show the pRR, EM and F1 scores respectively for each data split. Scores for QRCD training data, validation data and testing data are represented in respective columns with headers 'Training', 'Development' and 'Test'. The best scores are in bold font.

Model	Training	Development	Test
R0	0.7803	0.5146	-
R1	**0.9739**	0.55809	-
R2(a)	0.9526	0.5711	-
R2(b)	0.4870	0.3872	-
R3	0.9721	0.5685	-
R4 (a)	0.9675	0.5829	**0.3075** (run03)
R4 (b)	0.9643	**0.5888**	0.2795 (run02)
R5	0.9287	0.9217	0.2873 (run01)

Table 2: Models R1-R5 pRR Scores for Train, Development and Test Data

Model	Training	Development	Test
R0	0.6394	0.2660	-
R1	**0.9478**	0.3211	-
R2(a)	0.8957	0.3394	-
R2(b)	0.2774	0.1284	-
R3	0.9352	0.3302	-
R4 (a)	0.9267	**0.3486**	**0.0882** (run03)
R4 (b)	0.9282	0.3119	0.0756 (run02)
R5	0.8690	0.7889	0.0756 (run01)

Table 3: Models R1-R5 Exact-Match Scores for Train, Development and Test Data

Model	Training	Development	Test
R0	0.7549	0.4834	-
R1	**0.9728**	0.5305	-
R2(a)	0.9509	0.5409	-
R2(b)	0.4501	0.3507	-
R3	0.9702	0.5276	-
R4 (a)	0.9657	0.5544	**0.2676** (run03)
R4 (b)	0.9629	**0.5677**	0.2465 (run02)
R5	0.9246	0.9213	0.2684 (run01)

Table 4: Models R1-R5 F1 Scores for Train, Development and Test Data

In the last model setting **R5** the training and development data are merged thereby giving best results on development data. Therefore, we deliberately do not highlight **R5** scores on development data as best. Only its results for the test data are taken into account. For development data, we select **R4(a)** giving best EM score and **R4(b)** giving best pRR and F1 scores.

We observed that the model **R1** fine-tuned over the BERT pre-trained model, asafaya/bert-base-arabic using Masked Language Model, secured a better pRR score of 55.80% on development data compared to two other models; model **R0** trained from scratch giving 51.46% pRR and the model **R2(b)** fine-tuned over bert-base-multilingual-uncased, pre-trained on 102 languages, giving 38.72% pRR.

We also note the fine-tuned model **R1** has strikingly higher scores, on all three metrics, for training data as compared to development data. This indicates its tendency towards over-fitting. We use the data augmentation scheme that curbs the problem to some extent, improving the results relatively in the models using this scheme i.e. **R2(a)**, **R3**, **R4(a)**, **R4(b)** and **R5**. However, during training the augmented data, sampled from AQQAC dataset, is not true representative of the development data and therefore its impact is small if not insignificant. Figure-3 shows the effect of data augmentation at the time of fine-tuning as compared to the model in Figure-2 fine-tuned on data without any augmentation.

We observed that the model **R2(a)** using data augmentation and fine-tuned over pre-trained BERT model for Arabic only out performs **R2(b)** also using data augmentation but fine-tuned over multi-lingual pre-trained BERT model. We used settings of **R2(a)** as base settings for later models R3 to R5 along with additional enhancements.

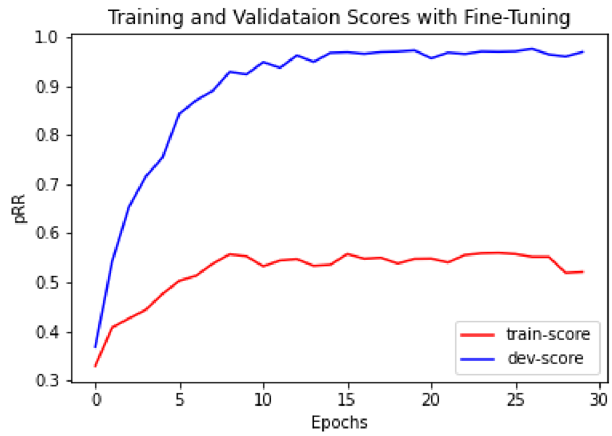

Figure 2: Model-R1 (Fine-tuned on bert-base-arabic) pRR Scores for Training and Development Data

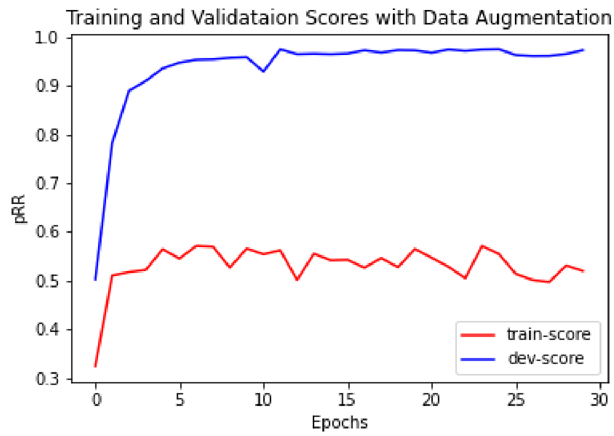

Figure 3: Model-R2(a) (Fine-tuned on bert-base-arabic with Augmented Training Data) pRR Scores for Training and Development Data

The model **R4** gave different results on two different training executions stated as **R4(a)** and **R4(b)**, which is due to permitting the shuffling of the training data for **R4(a)**. In both cases, we evaluated results after each epoch and saved the model at a checkpoint with best results. The execution of **R4(a)** shows best score of 30.75% pRR on test data, while **R5** stood second best with 28.73% pRR. Although, R5 included validation data for training, but its lack of showing improved results on test data could be attributed to one minor difference in saving the trained model state. Unlike other models, R5 was saved after 30 epochs and we did not evaluate it on validation data after every epoch to find the best checkpoint to save. This might have caused us to miss the best model state at some point during the course of epochs. Another reason could be that the augmented data size was not very significant.

Overall, in our approach the fine-tuning in combination with data augmentation technique and weight-decay value 0.01 generated the best scores of 58.88% pRR on development data and 30.75% on test data amongst all our settings and runs.

7. Conclusion and Future Work

We attempted to solve QuranQA shared task using BERT (Devlin et al., 2018) from scratch as well as fine-tuned over two different pre-trained variants. Moreover we opted for data augmentation and weight-decay regularization techniques to improve performance over the task.

Our key finding are thus summarized as follows:

- Fine-tuning over a pre-trained model specifically for Arabic language has leverage over the multi-lingual pre-trained model as well as training from scratch.
- Regularization methods like data augmentation and weight-decay enhance the performance by keeping the model from over-fitting.

In future work, we intend to apply following techniques in anticipation of improving the performance of our approach on QuranQA task.

- We expect to get enhanced performance by making architectural level changes in the model.
- We intend to increase the training data using techniques like back translation to generate rephrased questions or by replacing words in questions with synonyms.
- We intend to use different activation functions on hidden layers or even employ a different loss function that can help the model improve results to some extent.

8. Bibliographical References

Alqahtani, M. and Atwell, E. (2018a). Annotated corpus of arabic al-quran question and answer.

Alqahtani, M. M. and Atwell, E. (2018b). Developing bilingual arabic-english ontologies of al-quran. In *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, pages 96–101. IEEE.

Atef, A., Mattar, B., Sherif, S., Elrefai, E., and Torki, M. (2020). Aqad: 17,000+ arabic questions for machine comprehension of text. In *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–6. IEEE.

Banko, M., Brill, E., Dumais, S., and Lin, J. (2002). Askmsr: Question answering using the worldwide web. In *Proceedings of 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, pages 7–9.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Carrino, C. P., Costa-jussà, M. R., and Fonollosa, J. A. (2019). Automatic spanish translation of the squad dataset for multilingual question answering. *arXiv preprint arXiv:1912.05200*.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Chen, J., Lin, S.-t., and Durrett, G. (2019). Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*.

Chen, D. (2018). *Neural reading comprehension and beyond*. Stanford University.

Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., and Palomaki, J. (2020). Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Darvishi, K., Shahbodagh, N., Abbasiantaeb, Z., and Momtazi, S. (2022). Pquad: A persian question answering dataset. *arXiv preprint arXiv:2202.06219*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Falconer, J. (2011). Google: Our new search strategy is to compute answers, not links. `https://thenextweb.com/news/google-our-new-search-strategy-is-to-compute-a` [Online; accessed 10-May-2022].

Ismail, W. S. and Homsi, M. N. (2018). Dawqas: A dataset for arabic why question answering system. *Procedia computer science*, 142:123–131.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. (2019). Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Lee, K., Yoon, K., Park, S., and Hwang, S.-w. (2018). Semi-supervised training data generation for multilingual question answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Malhas, R. and Elsayed, T. (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)*, 19(6):1–21.

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 Pro-*

cessing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022).

Mozannar, H., Hajal, K. E., Maamary, E., and Hajj, H. (2019). Neural arabic question answering. *arXiv preprint arXiv:1906.05394*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Ruder, S. (2021). Multi-domain multilingual question answering. `https://ruder.io/multi-qa-tutorial/index.html#open-retrieval-qa-vs-reading-comprehension`. [Online; accessed 25-April-2022].

Safaya, A., Abdullatif, M., and Yuret, D. (2020). Kuisail 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.

Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. (2018). Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., and Le, Q. V. (2018). Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.