Proceedings of the OSACT 2022 Workshop @LREC2022, pages 154–161
Marseille, 20 June 2022

TCE* at Qur'an QA 2022: Arabic Language Question Answering Over Holy Qur'an Using a Post-Processed Ensemble of BERT-based Models

Mohammed ElKomy, Amany M. Sarhan
Computer and Control Engineering Department, Faculty of Engineering, Tanta University.
{mohammed.a.elkomy, amany_sarhan}@f-eng.tanta.edu.eg

Abstract

In recent years, we witnessed great progress in different tasks of natural language understanding using machine learning. Question answering is one of these tasks which is used by search engines and social media platforms for improved user experience. Arabic is the language of the Holy Qur'an; the sacred text for 1.8 billion people across the world. Arabic is a challenging language for Natural Language Processing (NLP) due to its complex structures. In this article, we describe our attempts at OSACT5 Qur'an QA 2022 Shared Task, which is a question answering challenge on the Holy Qur'an in Arabic. We propose an ensemble learning model based on Arabic variants of BERT models. In addition, we perform post-processing to enhance the model predictions. Our system achieves a Partial Reciprocal Rank (pRR) score of 56.6% on the official test set.

Keywords: Natural Language Processing, Extractive Question Answering, Holy Qur'an Computational Linguistics, Arabic SharedTask, Ensemble Bert

1. Introduction

Nowadays, the web and social media are integral parts of our modern digital life as they are the main sources of the unprecedented amounts of data we have. Thanks to the breakthrough in deep learning, search engines are no longer restricted to keyword matching; instead, they are currently able to understand queries in natural language and satisfy the intended information need of users. Question Answering (QA) is an essential task in information retrieval which is forming the basis for the new frontier of search engines. Plenty of studies on question answering systems has been performed on English and other languages. However, very few attempts have addressed the problem of Arabic question answering (Alwaneen et al., 2022).

Arabic NLP research in question answering is particularly challenging due to the scarcity of resources and the lack of processing tools available for Arabic. Arabic language, as well, has some unique characteristics by being a highly inflectional and derivational language with complex morphological structures (Abdelnasser et al., 2014). The Holy Qur'an is the sacred text for Muslims around the globe and it is the main source for teachings and legislation in Islam (Malhas and Elsayed, 2020), there are 114 chapters in Qur'an corresponding to 6,236 verses, every verse consists of a sequence of words in Classical Arabic (CA) dating back to 1400 years ago.

This paper describes our proposed solutions for OSACT5 Qur'an QA 2022 shared task. The shared task introduced QRCD (The Qur'anic Reading Comprehension Dataset), which is a dataset for extractive question answering. First, we experimented with a variety of Arabic pre-trained Bidirectional Encoder Representations from Transformers (BERT) models, then we implemented an Ensemble approach to get more robust results from a mixture of experts (MOEs). After that, we propose some post-processing operations to enhance the quality of answers according to the official evaluation measures. The task is evaluated as a ranking task according to the Partial Reciprocal Rank (pRR) metric.

The rest of this paper is organized as follows. In section 2, we describe the related work, in section 3, we outline the dataset details and official evaluation measures, in section 4, we explain the system design and the implementation details, in section 5, we report the system evaluation results, and finally section 6 concludes the paper. [1]

2. Related Work

Question answering systems have been an active point of research in recent years, particularly for highly-resourced languages such as English. (Rajpurkar et al., 2016) introduced SQuAD1.0 dataset which is a widely used dataset for question answering in English. To tackle the data scarcity in Arabic NLP, (Mozannar et al., 2019) presented Arabic Reading Comprehension Dataset (ARCD) which consists of 1,395 questions posed by crowdworkers. Moreover, (Mozannar et al., 2019) automatically translated SQuAD1.0 using google translation services. Only a little attention has been paid to question answering on Qur'an. (Abdel-

[1] The source code and trained models are available at https://github.com/mohammed-elkomy/quran-qa.

[2] To enable fair comparison among the teams, the organizers only considered 238 examples for the official test-split results and excluded 36 samples due to being very similar to public splits.

[*] Tanta Computer Engineering

nasser et al., 2014) proposed a Support Vector Machine (SVM) question answering system with handcrafted features to extract answers from both Qur'an and its interpretation books (Tafseer). Recent work by (Malhas and Elsayed, 2020) introduced `AyaTEC` as the first fully reusable test collection for Arabic QA on the Holy Qur'an where questions are posed in Modern Standard Arabic (MSA) and their corresponding answers are qur'anic verses in CA.

3. Dataset and Task Description

In this section, we describe QRCD (The Qur'anic Reading Comprehension Dataset), the problem definition and official evaluation metrics of the Qur'an QA 2022 shared task of answering questions on the holy Qur'an (Malhas et al., 2022b).

3.1. Dataset Details

The Qur'anic Reading Comprehension Dataset (QRCD) (Malhas et al., 2022a) is the first large scale question answering dataset on the holy Qur'an text. It was introduced as a part of the Qur'an QA 2022 Shared Task for question answering (Malhas et al., 2022a). The dataset consists of 1,093 tuples of question-passage pairs that are coupled with their extractive answers to constitute 1,337 question-passage-answer triplets. The question-passage pairs are split into training, development and testing sets as shown in Table 1. The dataset follows the same format as the commonly used reading comprehension dataset SQuAD1.0 (Rajpurkar et al., 2016). However, QRCD is quite different in terms of size as it is much smaller than SQuAD1.0 which contains 100k unique pairs/questions. In addition, unlike SQuAD, the QCRD contains a small number of unique questions, each of which is repeated multiple times with different passage and answer pairs. As shown in Table 1, the number of unique questions in QRCD is much lower than the number of question-passage pairs. This poses an additional challenge for learning a question answering system which should be able to predict different answers to the same question under different passage contexts.

Split Aspect	**Train**	**Dev**	**Test**
Question-passage pairs	710	109	274[2]
Unique questions	118	17	34

Table 1: Number of question-passage pairs and number of unique questions in each split of QRCD dataset.

The QRCD dataset draws its inspiration from the prior work `AyaTEC` by reformulating the test collection into an extractive question answering task (Malhas et al., 2022a). Each sample in QRCD is a question-passage-answer triplet which comprises a question in MSA, a passage taken from the Holy Qur'an [3] that spans one or more consecutive verses, and an answer to the question extracted from the passage. Figure 1 demonstrates an example from the QRCD dataset[4].

3.2. Qur'an QA Shared Task Description

The Qur'an QA 2022 shared task (Malhas et al., 2022a) aims to develop models for extractive question answering on the holy Qur'an passages. Given a Qura'nic passage and a question, the solution to the shared task should extract the answer to the question from the input passage. The answer always exists as a span within the given passage. Questions could be either factoid or non-factoid. Solutions to the shared task are required to extract any correct answer to the input question from the context passage even when the passage has more than one answer.

3.3. Task Evaluation Measures

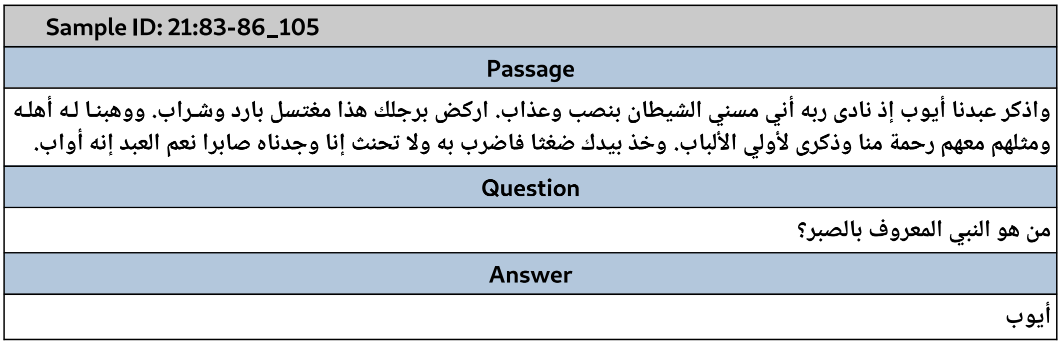
This question answering task is evaluated as a *ranking* task. The QA system will return up to 5 potential answers ranked from the best to the worst according to their probability of correctness. The task evaluation measure produces a higher score when the correct answer is ranked at a higher position. When the correct answer is predicted among the 5 potential answers but it is at a lower rank, then the evaluation score is discounted. The task adopts the *partial Reciprocal Rank (**pRR**)* (Malhas and Elsayed, 2020) as the official evaluation metric. Partial Reciprocal Rank (**pRR**) is a variant of the *Reciprocal Rank* (RR) evaluation metric which is a commonly used metric for ranking tasks. Unlike *Reciprocal Rank* (RR), the *partial Reciprocal Rank (**pRR**)* will give credit to systems that predict answers with a partial inexact matching with the correct answer. Equation 1 formally describes the **pRR** metric evaluation for a ranked list of answers A, where m_{r_k} is the partial matching score for the returned answer at the k^{th} rank, as k is taken to be equal to the rank position of the first answer with a non-zero matching score. More details can be found at (Malhas and Elsayed, 2020) 5.2.

$$pRR(A) = \frac{m_{r_k}}{k}; k = \min\{k \mid m_{r_k} > 0\} \tag{1}$$

In addition to the pRR scores, the task evaluation system reports other metrics such as the Exact Match (**EM**) and **F1@1**. The **EM** score is a binary measure that will be equal to one when the top predicted answer exactly matches the ground truth answer. The **F1@1** metric measures the degree of token overlap between the top predicted answer and any of the ground truth

[3] The Holy Qur'an is a very special classical Arabic text revealed 1,400 years ago, making it extremely challenging for computational linguistics tasks.

[4] The verses from the Holy Qur'an in the dataset come from the simple-clean text style (diacritics removed) from Tanzil Project.



Sample ID: 21:83-86_105

Passage

واذكر عبدنا أيوب إذ نادى ربه أني مسني الشيطان بنصب وعذاب. اركض برجلك هذا مغتسل بارد وشراب. ووهبنا له أهله
ومثلهم معهم رحمة منا وذكرى لأولي الألباب. وخذ بيدك ضغثا فاضرب به ولا تحنث إنا وجدناه صابرا نعم العبد إنه أواب.

Question

من هو النبي المعروف بالصبر؟

Answer

أيوب

Figure 1: An example of question-passage-answer triplet from QRCD (Malhas et al., 2022a).

answers. Scores computed from individual passage-question-answer triplets are averaged to compute the overall score over the entire evaluation dataset.

4. QA System Design

Bidirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) (Devlin et al., 2019) models achieve state-of-the-art results in many natural language understanding problems. Our solution is an ensemble of BERT based models pre-trained on Arabic language corpora and fine-tuned on the shared task dataset. We built an ensemble that merges predictions from individual models. Additionally, we also designed and implemented a set of post-processing operations that aim to improve the quality of predicted answers and boost the task evaluation measure. In this section, we describe our QA system developed to solve the Qur'an QA 2022 challenge. First, we give a brief background on BERT models and their usage for question answering tasks. Then we provide an overview of Arabic language BERT models that we used to build our ensemble. Finally, we provide the details of our ensemble building approach and the proposed post-processing operations to improve predicted answers quality.

4.1. BERT for Question Answering

4.1.1. BERT

BERT models achieve their state-of-the-art performance due to a procedure called pre-training which allows BERT to discover the language structures and patterns. BERT uses two pre-training tasks, namely masked language model (MLM) and next sentence prediction (NSP) (Devlin et al., 2019). After that, a second stage called fine-tuning, which is performed to adapt the model for a downstream task by making use of the features learnt during the pre-training phase.

4.1.2. Question Answering using BERT

As mentioned in 4.1.1, the fine-tuning phase takes a pre-trained model and stacks a randomly-initialized output layer suitable for a particular downstream task. For extractive question answering, both the question and passage are tokenized and packed into a single sequence and the output layer is required to give a probability for the ith token in the passage to be the start of the answer span P_i using the dot product of a start vector S and the ith token's hidden representation T_i as seen from equation 2, a similar analysis holds for the end of the answer span with an end vector E. The score of a candidate's answer span from the ith token to the jth token is defined in Equation 3. Answers are only accepted for $j \geq i$ since the two probability distributions are independent and not guaranteed to produce a valid span (Devlin et al., 2019). S and E are trainable weights and randomly initialized layers stacked on top of pre-trained BERT. In our case the system is not limited to just one answer as in SQuAD1.0 (Rajpurkar et al., 2016), instead, a ranked list of 20 answers is generated from the model and ranked based on the span score as in Equation 3.

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}} \tag{2}$$

$$Span_{i,j} = ST_i + ET_j \tag{3}$$

4.2. Arabic variants of BERT model

The standard BERT model variants are not pre-trained on Arabic text which hinders the development of Arabic NLP. Nevertheless, various researchers working on the Arabic natural language understanding have developed variants of BERT models that were trained on Arabic corpora. We made use of those models, which are discussed later in this section, to build our ensemble.

4.2.1. AraBERT

The work by (Antoun et al., 2020) introduced AraBERT model which inherits the exact same architecture from BERT. However, being pre-trained on a large Arabic corpus of 24GB of text collected from news articles and Wikipedia dumps. In this work, we use **bert-large-arabertv02** and **bert-base-arabertv02** available on the huggingface community (Wolf et al., 2019).

4.2.2. QARiB

QARiB (Abdelali et al., 2021) is another Arabic BERT variant pretrained on a mixture of formal and informal

Arabic with state-of-the-art support for Arabic dialects and social media text, (Abdelali et al., 2021) released 5 BERT models to the community pre-trained on corpora of different sizes.

4.2.3. ARBERT and MARBERT

ARBERT and MARBERT (Abdul-Mageed et al., 2021) are two Arabic-specific Transformer-based MLM pre-trained on a widely large Modern Standard Arabic (MSA) corpus of 61GB of text for the case of ARBERT, while MARBERT is pre-trained on 128GB of text focused on both dialectal Arabic (DA) and MSA.

4.3. Training Details

We fine-tune a set of five Arabic BERT models as mentioned in 4.2, namely AraBERT-v02$_{\text{Large}}$[5], AraBERT-v02$_{\text{Base}}$, QARiB$_{\text{Base}}$, ARBERT and MARBERT [6].
The training objective is to maximize the log-likelihoods of the correct start and end token positions (Devlin et al., 2019). For the development phase, we train BERT$_{\text{Base}}$ and BERT$_{\text{Large}}$ for 50 and 65 epochs respectively, looking for the epoch at which the model performs best on the validation split. For the test phase, we train BERT$_{\text{Base}}$ and BERT$_{\text{Large}}$ for 32 and 40 respectively. We used a batch size of 8 for BERT$_{\text{Large}}$ and 16 for BERT$_{\text{Base}}$ and a learning rate of 2e-5 for all of our models.

4.4. Span Voting Ensemble

In this work, we build an ensemble from several different Arabic BERT models that we discussed earlier in 4.3. The ensemble approach is effective for cancelling the noise exhibited by individual models through majority voting among experts. i.e. Mixture of Experts (MOE). We treat the answer spans as discrete entities. For each sample, we consider the top 20 predictions made by each model along with its correctness probability. For each candidate prediction, we compute the sum of its associated correctness probabilities from all models. We formulate the voting process as follows.

$$\alpha_{s,e} = \sum_{j=0}^{M} \alpha_{s,e}^{j}$$

Where $\alpha_{s,e}$ represents the summed ensemble correctness probability for the answer span starting at token s and ending at token e, and $\alpha_{s,e}^{j}$ represents the correctness probability for the same answer span for the j^{th} expert of the M experts taken into account.
After that, the set of all possible answers considered by the ensemble is sorted according to their summed ensemble correctness probabilities. Finally, the entire ranked list is post-processed and truncated for only the top 5 answers to be evaluated by **pRR@5**.

4.5. Post-processing

By carefully reviewing the answer spans predicted by the BERT models we fine-tuned, we found some systematic errors causing sub-optimal predictions. Here we propose some basic post-processing rules to improve the model predictions. The post-processing pipeline takes a ranked list of answer spans, it typically takes at least 20 answer spans from a single model or the span-voting ensemble. Figure 2 provides an illustrative example of the post-processing pipeline. Due to the limited space, we only consider the top 15 answer spans from the original system outputs.

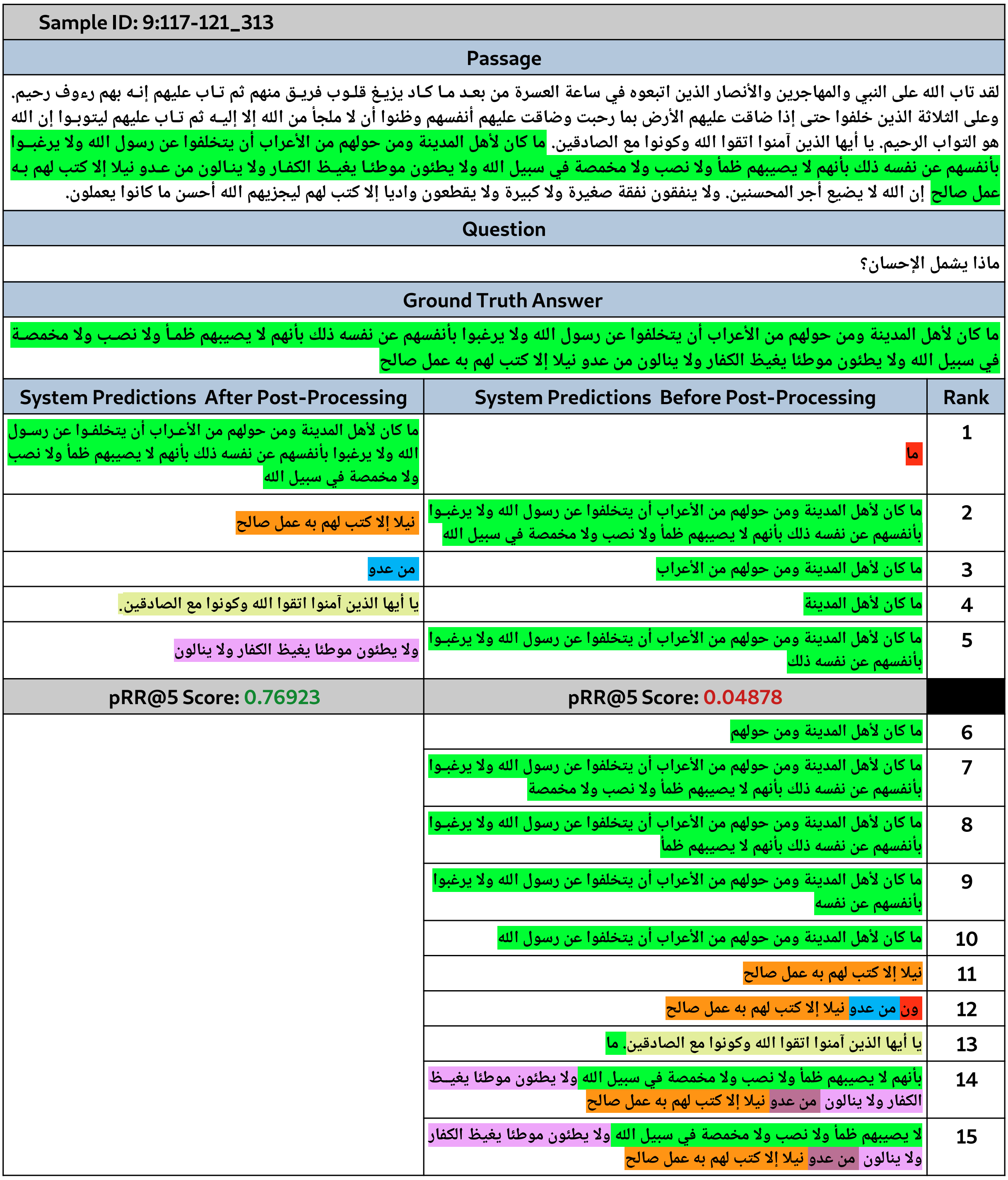
4.5.1. Handling Sub-words

Before feeding the input to BERT, it must undergo the tokenization step. Tokenization is the process of splitting a sentence into tokens. For BERT, WordPiece tokenizer is commonly used as a subword tokenizer. This makes the system susceptible to producing an output with incomplete words like ”وفي الرقاب والغارم” which will be penalized by the evaluation process, we perform a simple post-processing rule to extend or drop tokens such that we do not have broken words and this simple rule produces a corresponding output ”وفي الرقاب والغارمين”, for the previously mentioned example. In Figure 2, we dropped the sub-token ”ون” at rank 12 which is part of ”ينالون”.

4.5.2. Redundancy Elimination

We analyzed the predictions of a variety of our fine-tuned models and discovered that most of the answer spans are highly overlapping with each other, making the ranked list suboptimal with respect to the pRR metric in 3.3. The rationale behind this is, the pRR metric *only* considers the (k+1)th prediction when there is no overlap with any of the ground-truth answer tokens for the kth prediction, This implies repeating any of the words from the kth prediction in the (k+1)th prediction is suboptimal, which is a common behaviour exhibited by BERT for QA. Here we present algorithm 1 which ensures the elimination of span overlap among the answers of a ranked list returned by the system. An illustrative example showing the predictions before and after the application of this rule is given in Figure 2, the colours used for highlighting text depict the high overlap between unprocessed answers, it clearly shows the span ”ما كان لأهل المدينة” is common in the first few unprocessed answers. After applying this rule, the ranked list after post-processing better covers the text

[5] A subscript Large and Base refers to the model size.
[6] ARBERT and MARBERT uses BERT$_{\text{Base}}$ architecture.



Sample ID: 9:117-121_313		
Passage		
لقد تاب الله على النبي والمهاجرين والأنصار الذين اتبعوه في ساعة العسرة من بعد ما كاد يزيغ قلوب فريق منهم ثم تاب عليهم إنه بهم رءوف رحيم. وعلى الثلاثة الذين خلفوا حتى إذا ضاقت عليهم الأرض بما رحبت وضاقت عليهم أنفسهم وظنوا أن لا ملجأ من الله إلا إليه ثم تاب عليهم ليتوبوا إن الله هو التواب الرحيم. يا أيها الذين آمنوا اتقوا الله وكونوا مع الصادقين. ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله ولا يرغبوا بأنفسهم عن نفسه ذلك بأنهم لا يصيبهم ظمأ ولا نصب ولا مخمصة في سبيل الله ولا يطئون موطئا يغيظ الكفار ولا ينالون من عدو نيلا إلا كتب لهم به عمل صالح إن الله لا يضيع أجر المحسنين. ولا ينفقون نفقة صغيرة ولا كبيرة ولا يقطعون واديا إلا كتب لهم ليجزيهم الله أحسن ما كانوا يعملون.		
Question		
ماذا يشمل الإحسان؟		
Ground Truth Answer		
ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله ولا يرغبوا بأنفسهم عن نفسه ذلك بأنهم لا يصيبهم ظمأ ولا نصب ولا مخمصة في سبيل الله ولا يطئون موطئا يغيظ الكفار ولا ينالون من عدو نيلا إلا كتب لهم به عمل صالح		
System Predictions After Post-Processing	**System Predictions Before Post-Processing**	**Rank**
ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله ولا يرغبوا بأنفسهم عن نفسه ذلك بأنهم لا يصيبهم ظمأ ولا نصب ولا مخمصة في سبيل الله	ما	1
نيلا إلا كتب لهم به عمل صالح	ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله ولا يرغبوا بأنفسهم عن نفسه ذلك بأنهم لا يصيبهم ظمأ ولا نصب ولا مخمصة في سبيل الله	2
من عدو	ما كان لأهل المدينة ومن حولهم من الأعراب	3
يا أيها الذين آمنوا اتقوا الله وكونوا مع الصادقين.	ما كان لأهل المدينة	4
ولا يطئون موطئا يغيظ الكفار ولا ينالون	ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله ولا يرغبوا بأنفسهم عن نفسه ذلك	5
pRR@5 Score: 0.76923	pRR@5 Score: 0.04878	
	ما كان لأهل المدينة ومن حولهم	6
	ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله ولا يرغبوا بأنفسهم عن نفسه ذلك بأنهم لا يصيبهم ظمأ ولا نصب ولا مخمصة	7
	ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله ولا يرغبوا بأنفسهم عن نفسه ذلك بأنهم لا يصيبهم ظمأ	8
	ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله ولا يرغبوا بأنفسهم عن نفسه	9
	ما كان لأهل المدينة ومن حولهم من الأعراب أن يتخلفوا عن رسول الله	10
	نيلا إلا كتب لهم به عمل صالح	11
	ون من عدو نيلا إلا كتب لهم به عمل صالح	12
	يا أيها الذين آمنوا اتقوا الله وكونوا مع الصادقين. ما	13
	بأنهم لا يصيبهم ظمأ ولا نصب ولا مخمصة في سبيل الله ولا يطئون موطئا يغيظ الكفار ولا ينالون من عدو نيلا إلا كتب لهم به عمل صالح	14
	لا يصيبهم ظمأ ولا نصب ولا مخمصة في سبيل الله ولا يطئون موطئا يغيظ الكفار ولا ينالون من عدو نيلا إلا كتب لهم به عمل صالح	15

Figure 2: A comprehensive example for post-processing. Here we present the outputs of the system before post-processing (original outputs) on the right and after post-processing on the left. We used green to highlight the ground truth answer or parts of it extracted by the system. Words and sub-words marked by red are dropped according to the rules 4.5.1 and 4.5.3. Other colours used for highlighting text depict the high overlap among the predictions before post-processing.

in the passage and the pRR score increases from 0.048 to 0.769 after post-processing. Figure 3 shows the distribution of the per-sample pRR score before and after post-processing, the percentage of development samples of the first two bins after post-processing is reduced, which means we are less observing completely wrong answers after post-processing.

4.5.3. Uninformative Answer Removal

In this post-processing rule, we remove the uninformative answers from the ranked list, We define an

Algorithm 1 Redundancy Elimination Algorithm

```
Input: $P$, passage text; $A$, input answers list.
Output: post-processed answers list.
$N_P \leftarrow$ Number of words in $P$.
$M_{seen} \leftarrow 0_{N_P}$                      ▷ Initialized to Zero
                                 ▷ The mask used to track seen words
$A_{post} \leftarrow$: []          ▷ Output Initialized to an empty list

for each $a$ in $A$ do
    $s \leftarrow$ get start-word index of span $a$.
    $e \leftarrow$ get end-word index of span $a$.
    $a_{seen} \leftarrow M_{seen}[s{:}e]$
                                 ▷ Get seen slice of answer span $a$
    if $a_{seen}$ has any zero then
                                 ▷ at least a word is not seen,
                                 ▷ marked by 1 in $a_{seen}$
        $a_{\text{unseen of } P} \leftarrow$ get_unseen_seqs($a_{seen}$,$P$)
    ▷ brings unseen contiguous sequences of words.
        for each $seq_{\text{unseen}}$ in $a_{\text{unseen of } P}$ do
                                 ▷ $seq_{\text{unseen}}$ is a subsequence of words
                                 ▷ with $a_{seen}$ consisting of only zeros
            $s_{unseen} \leftarrow$ start-word index of $seq_{\text{unseen}}$.
            $e_{unseen} \leftarrow$ end-word index of $seq_{\text{unseen}}$.
            $seq_{text} \leftarrow$ text spanned by $seq_{\text{unseen}}$.
            $M_{seen}[s_{unseen}{:}\ e_{unseen}] = 1$
                                 ▷ Mark tokens as seen
            $A_{post} = A_{post} \smile seq_{text}$
    ▷ Append this unseen part of answer span $a$
        end for

    end if
end for
```

uninformative answer as having one of the following conditions:

1. All of the stemmed answer tokens exist in the stemmed question tokens, for example, a question like ”ما هي شجرة الزقوم؟” with a complete answer span predicted by the system ”الزقوم” is considered an uninformative answer.

2. The whole answer span consists of stop-words, which can never meet the information need of a question in QRCD, for example, answer spans like ”اذا”, ”ليس”, ” ثم” are considered uninformative answers.

Uninformative answers come from two sources, first, the original output of the BERT model without post-processing and second, the post-processed outputs after removing redundant tokens as in 4.5.2. For the example in Figure 2, the answer at rank 1 ”لما” is rejected due to being uninformative.

4.5.4. Updating the Ranked List

After performing the post-processing pipeline described in 4.5.2, 4.5.3 and 4.5.1 in order, we may end up with a new ranked list with more than 5 answer spans, we only consider the top 5 answer spans in the post-processed ranked list for the metric evaluation (pRR@5).

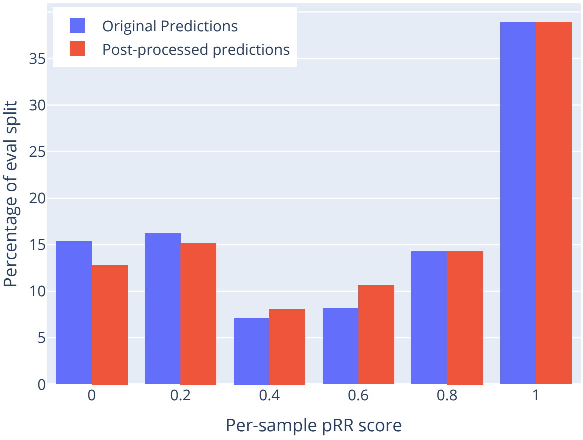


Figure 3: The post-processing impact on the per-sample pRR score distribution, for this plot, we used the ouputs of the models marked with † in Table 2.

5. Experimental Evaluation

In this section, we report our trained models' results along with the official scores and run details on the Codalab competition.

5.1. Development Phase

In this phase, we did not have access to the test dataset. We trained our models on the training set as in 4.3 while only saving the best performing model on the validation split. We observed a large variation (around $\pm3\%$) in the pRR score reported for the same starting checkpoint with different seeds, we relate this to the small size of the validation split. To enable fair comparison, we average the reported scores for the same model trained multiple times with different seeds as shown in Table 2. For the ensemble method, we considered 15 checkpoints with different seeds for each of AraBERT-v02$_{\text{Large}}$, AraBERT-v02$_{\text{Base}}$ and ARBERT (marked with † in the table), adding up to 45 experts, labelled as **Ensemble**$_{\textbf{Vanilla}}$ in Table 2. After that, we performed the post-processing step on the ensemble outputs referred to as **Ensemble**$_{\textbf{POST}}$ in Table 2. Also from the table, QARiB$_{\text{Base}}$ and MARBERT are performing worse on average, because of being primarily targeted for dialectal Arabic and social media text.

Single Models	**EM** (%)	**F1** (%)	**pRR** (%)
†arabertv02$_{\text{Large}}$	37.2	59.0	**61.7**
†arabertv02$_{\text{Base}}$	36.5	58.5	60.7
†ARBERT	37.3	58.7	60.9
MARBERT	32.1	51.5	53.9
QARiB$_{\text{Base}}$	25.9	45.3	48.1

Ensemble	**EM** (%)	**F1** (%)	**pRR** (%)
Ensemble$_{\textbf{Vanilla}}$	39.4	59.4	63.97
Ensemble$_{\textbf{POST}}$	38.5	59.4	**65.22**

Table 2: QRCD development split results, reported metrics for single models are averaged for a number of model checkpoints trained with different seeds, while for the ensemble case, it is a single instance produced from combining different checkpoints. **Ensemble$_{\textbf{Vanilla}}$** refers to combining 45 checkpoints of models indicated by † (15 for each). **Ensemble$_{\textbf{POST}}$** represents the **Ensemble$_{\textbf{Vanilla}}$** output after post-processing.

Run ID	**EM** (%)	**F1** (%)	**pRR** (%)
Ensemble$_{\text{keep}}$	26.8	48.5	55.7
Ensemble$_{\text{remove}}$	26.8	50.0	**56.6**

Table 3: Test phase official results on QRCD dataset, Each ensemble reported is a span-voting ensemble combining all models in Table 4. "keep" subscript refers to keeping uninformative answer spans as discussed in 4.5.3, on the other hand, "remove" subscript points to removing uninformative answer spans.

5.2. Competition Final Testing Phase

During the competition's final testing phase, we had access to the test dataset without labels, and participants were required to submit at most 3 submissions produced by their proposed systems. We trained our models on both the training and development datasets as in 4.3. The trained models were combined in an ensemble as discussed in 4.4, then we performed the post-processing. Table 4 shows the number of models used as experts for span-voting ensemble in the test phase. In Table 3, we outline the official results obtained for our submissions. All of them are ensemble-based due to the significant variations we observed in the development phase. Combining **all** of the models in Table 4 followed by post-processing the output predictions with uninformative span **removal** performs best, this is marked in the table as **Ensemble$_{\textbf{remove}}$**. There is a significant gap between the results in the development and test phases as in tables 2 and 3 respectively, we relate this to the small size of the validation split against the test split, another reason is excluding 36 samples from the test split since their questions were similar to the public splits as indicated by the organizers in the official test phase results.

Models	**Num**
arabertv02$_{\text{Large}}$	16
arabertv02$_{\text{Base}}$	18
ARBERT	17

Table 4: Number of models involved in the final ensemble of the test phase.

6. Conclusion and Future Work

In this work, we leveraged the pre-trained Arabic language models to solve the Qur'an QA 2022 Shared Task. We fine-tuned a variety of BERT models optimized for the Arabic language. We proposed some post-processing operations to enhance the quality of answers aligning with the official measure. Ensemble-based approaches are effective to produce more robust predictions. In the future, we will further study how to incorporate a stacking ensemble approach with multiple stages to achieve better performance rather than a voting ensemble as used in this study. We will also investigate why we observed huge variations in the reported results by performing extensive cross-validation.

7. Acknowledgements

We appreciate the efforts and assistance of Dr Moustafa Alzantot regarding the paper-writing phase, and recommendations during the implementation.

8. Bibliographical References

Abdelali, A., Hassan, S., Mubarak, H., Darwish, K., and Samih, Y. (2021). Pre-training bert on arabic tweets: Practical considerations.

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.

Abdul-Mageed, M., Elmadany, A., and Nagoudi, E. M. B. (2021). ARBERT & MARBERT: Deep bidirectional transformers for Arabic. 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 7088–7105, Online, August. Association for Computational Linguistics.

Alwaneen, T. H., Azmi, A. M., Aboalsamh, H. A., Cambria, E., and Hussain, A. (2022). Arabic question answering system: a survey. *Artificial Intelligence Review*, 55(1):207–253, Jan.

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.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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.

Malhas, R., Mansour, W., and Elsayed, T. (2022a). 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)*.

Malhas, R., Mansour, W., and Elsayed, T. (2022b). Qur'an qa 2022 shared task. `https://gitlab.com/bigirqu/quranqa`.

Mozannar, H., Maamary, E., El Hajal, K., and Hajj, H. (2019). Neural Arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118, Florence, Italy, August. Association for Computational Linguistics.

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.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.