Al-Jawaab at Qur'an QA 2023 Shared Task: Exploring Embeddings and GPT Models for Passage Retrieval and Reading Comprehension

Abdulrezzak Zekiye
Istanbul, Türkiye
abdalrazak.zk@gmail.com

Fadi Amroush
Niuversity / Berlin, Germany
fadi.amr@niuversity.com

Abstract

This paper introduces a comprehensive system designed to address two natural language processing tasks: Passage Retrieval (Task A) and Reading Comprehension (Task B), applied to datasets related to the Holy Qur'an. Task A was treated as a measurement of a textual similarity problem where the system leverages OpenAI's "text-embedding-ada-002" embedding model to transform textual content into numerical representations, with cosine similarity serving as the proximity metric. Task B focuses on the extraction of answers from Qur'anic passages, employing the Generative Pre-trained Transformer-4 (GPT-4) language model. In Task A, the system is evaluated using the Mean Average Precision (MAP) metric, achieving MAP scores of 0.109438 and 0.06426543057 on the development and test datasets with an optimal similarity threshold set at 0.85. Task B evaluation employs partial Average Precision (pAP), where our system surpasses a baseline whole-passage retriever with pAP scores of 0.470 and 0.5393130538 on the development and test datasets, respectively.

Holy Qur'an, passage retrieval, reading comprehensive, GPT-4, embeddings

1 Introduction

Establishing a dependable method for providing accurate responses and citing relevant passages from the Holy Qur'an within the framework of natural language processing represents a crucial and challenging endeavor. The creation of a reliable model capable of delivering precise answers to inquiries about Islam and the Holy Qur'an holds substantial potential. It not only serves as a valuable resource for facilitating accurate information retrieval but also as a potent tool for automatically detecting and countering the dissemination of false information on the internet and social media platforms. Qur'an QA 2023 Shared Task (Malhas et al., 2023) encourages researchers to work on two important tasks, Task A: Passage Retrieval and Task B: Reading Comprehension.

Task A: Passage Retrieval. This task involves providing a ranked list of passages from the Holy Qur'an that potentially contain answers to a given free-text question in Modern Standard Arabic (MSA). The task encompasses both factoid and non-factoid questions where factoid questions have short answers such as names and numerical values, and non-factoid questions need explanations, reasoning, or opinions to provide an answer (Surdeanu et al., 2011). This task also includes certain questions within the dataset that lack corresponding answers in the Holy Qur'an. The system should return a ranked list containing up to 10 Qur'anic passages believed to contain the answer(s) to the given question if any, and "no answers." in case there is no answer from the Holy Qur'an.

Task B: Reading Comprehension. The task involves working with a particular Qur'anic passage, which comprises consecutive verses from a specific Surah in the Holy Qur'an, along with a free-text question presented in MSA pertaining to that passage. The primary objective is for a system to identify and extract all answers to a question that are explicitly mentioned within the corresponding passage. This approach differs from the previous Qur'an QA 2022 task, where the system was required to return any answer (Malhas et al., 2022). These answers are expected to be contiguous spans of text within the passage. Similar to Task A, the questions themselves can encompass both factoid and non-factoid types and the system should return up to 10 answers out of the provided passage, or an empty set representing a "no answer" case.

The structure of this paper is as follows: In Section 2, we provide an overview of the datasets employed for Tasks A and B. Section 3 details the methodologies utilized to address both tasks. Our results are presented in Section 4. Lastly, Section 5 offers a discussion of the results.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 743–747
December 7, 2023 ©2023 Association for Computational Linguistics

2 Data

2.1 Task A: Passage Retrieval

Task A dataset (Malhas et al., 2023; Malhas, 2023; Malhas and Elsayed, 2020; Swar, 2007) comprises three main components: the Qur'anic passage collection (QPC), the questions from the AyaTEC dataset, and query relevance judgements (QRels) as the assessments of how relevant these questions are to the passages within the QPC. The QPC was created by categorizing the 114 Qur'anic chapters, each of varying lengths, based on thematic divisions as outlined in the Thematic Holy Qur'an (Swar, 2007). This process led to a total of 1,266 distinct passages. The AyaTEC dataset has 199 questions and the QRels dataset consists of 1,132 gold Qur'anic passage-ids that are deemed relevant to each question. The output format of the system that solves task A should be as follows, where *tag* is used to indicate a human-readable model name: '<question-id>' Q0 '<passage-id>' '<rank>' '<relevance-score>' '<tag>'. The dataset was split as 70% for training, 10% for development, and 20% for testing, yielding 174 questions for the training, 25 for the development, and 52 for testing. From question-passage pairs point-of-view, the dataset had 972 pairs for training, 160 for development, and 427 for testing.

2.2 Task B: Reading Comprehension

In Task B, the used dataset is taken out of the Qur'anic Reading Comprehension Dataset (QRCD) v1.2 (Malhas et al., 2023, 2022; Malhas and Elsayed, 2022, 2020). QRCD v1.2 consists of 1,399 triplets of questions and corresponding passages, along with their extracted answers. The questions with "no answer" constitute 15% of the questions in the QRCD v1.2 dataset. The dataset was split as 64% for training, 10% for development, and 26% for testing. In other words, this task's dataset had 992 question-passage pairs for the training, 163 for the development, and 407 for testing.

3 System

3.1 Task A: Passage Retrieval

To solve this task, we measured the similarity between the question and all Qur'anic passages and then selected the most similar passages, up to 10. We put a threshold to indicate whether the question and a passage are similar or not. If no passage has a similarity score of more than the threshold, then a "no answer" case is indicated by the system. Similarity cannot be measured directly between two passages (the question and passage in our task). However, we can convert the passages to numerical representations and then measure the similarity between the resulting representations. Embedding models, such as BERT (Devlin et al., 2018), Word2Vec (Church, 2017), and GloVe (Pennington et al., 2014), can be used to convert a given text into a numerical space. In this work, we used OpenAI's embedding model which is called "text-embedding-ada-002" (OpenAI, 2023a). According to (OpenAI, 2023b), "text-embedding-ada-002" converts a given text into a 1536-dimension embedding vector with an 81.5% performance score on SenEval, a tool designed to assess the effectiveness of sentence embeddings (Conneau and Kiela, 2018). To measure the distance between two embedding vectors, we used the cosine similarity (Rahutomo et al., 2012).

3.2 Task B: Reading Comprehension

To solve this task, we utilized a handcrafted prompt with Generative Pre-trained Transformer-3 (GPT-3.5) and Generative Pre-trained Transformer-4 (GPT-4) language models in order to retrieve the answers to a question out of the corresponding passage, if any. GPT-3.5 is based on GPT-3 which is an autoregressive model with 175 billion parameters where it exhibits remarkable proficiency across a diverse range of natural language processing tasks (Brown et al., 2020). GPT-4 is a language model much larger than GPT-3.5 with about 1.7 trillion parameters (Schreiner, 2023). GPT-4 demonstrates performance comparable to that of humans with enhanced performance in terms of accuracy and adherence to desired behavioral criteria (Team, 2023).

In Task B, the system is supposed to return all the sections that contain an answer to a question out of a passage. While dealing with GPT models, we can think of the following scenarios:

1. **Scenario 1: Asking GPT model a direct question.** If we ask GPT-3.5 or GPT-4 to give us the answers to a question without a passage, it would provide us an answer where it might or might not be true, with a more accurate answer to be provided by GPT-4.

2. **Scenario 2: Asking GPT model a question with a passage to extract answers from.** Providing the passage to GPT and asking it to give

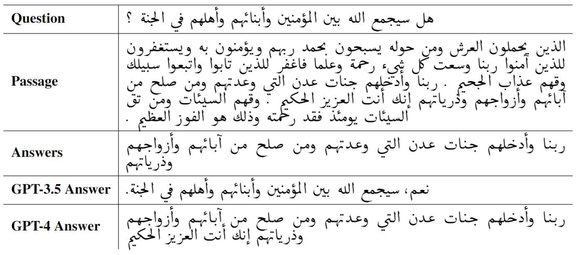
us answers to a question out of the provided passage would provide us with more reliable answers compared to scenario 1. However, there is still a chance for both models, GPT-3.5 and GPT-4, to provide us answers out of the provided passage.

3. **Scenario 3: The scenario is like scenario 2 but with making the model more determined.** When dealing with GPT-3.5 and GPT-4 models' APIs, we can control the *temperature* parameter to have lower values to get more determined answers. In other words, if we set this parameter to a value near zero, we will probably not get an answer out of the provided passage.

In our system, we followed the third scenario where we provided the GTP model with the prompt followed by the question and then the passage, along with setting the *temperature* parameter to zero. The *temperature* parameter varies between 0 and 2. Higher values yield more random output and lower values enhance the output determinism. The result of the model is not determined or fixed in every call where it sometimes returns an answer with double quotations, sometimes returned as a list with a special character in front of each answer, and so on. For that reason, we included a step that cleans the result by deleting special characters and white spaces out of the answer. The final step we have in the system is finding the corresponding start and end indices for each answer out of the passage as required by the task. If the provided answer is not in the passage, then we discard the answer since it means that the model has given an out-of-passage answer. We prompted the GPT model to return "no answer" in case the passage contains no answers to the provided question. As a result, our system returns "no answer" either if the GPT model gave a "no result" or all provided answers are out-of-passage. The prompt we used before is as follows:

ءجب على السؤال التالي من النص المرفق فقط .
لا تقم بإضافة أية شرح أو أية إجابة من خارج
النص. اكتب الاجابة أو الإجابات فقط، إن وجدت
أكثر من إجابة اكتبها على شكل تعدادات. الاجابة
يجب أن تكون فقط المقطع أو المقاطع التي تحوي
الجواب بدون أية زيادة. اجعل كل مقطع في سطر
منفصل. إن لم توجد إجابة، اكتب: "No Answer".

Fig. 1 shows an example from the dev dataset that consists of a question, a passage, and answers, along with the corresponding answers we obtained from GPT-3.5 and GPT-4.



Question	هل سيجمع الله بين المؤمنين وأبنائهم وأهلهم في الجنة ؟
Passage	الذين يحملون العرش ومن حوله يسبحون بحمد ربهم ويؤمنون به ويستغفرون للذين آمنوا ربنا وسعت كل شيء رحمة وعلما فاغفر للذين تابوا واتبعوا سبيلك وقهم عذاب الجحيم . ربنا وأدخلهم جنات عدن التي وعدتهم ومن صلح من آبائهم وأزواجهم وذرياتهم إنك أنت العزيز الحكيم . وقهم السيئات ومن تق السيئات يومئذ فقد رحمته وذلك هو الفوز العظيم .
Answers	ربنا وأدخلهم جنات عدن التي وعدتهم ومن صلح من آبائهم وأزواجهم وذرياتهم
GPT-3.5 Answer	نعم، سيجمع الله بين المؤمنين وأبنائهم وأهلهم في الجنة.
GPT-4 Answer	ربنا وأدخلهم جنات عدن التي وعدتهم ومن صلح من آبائهم وأزواجهم وذرياتهم إنك أنت العزيز الحكيم

Figure 1: Answers obtained from GPT-3.5 and GPT-4 for an example of Task B's dev dataset

4 Results

In this section, we present the results of our two models for Task A and Task B along with comparing them to the base model in each task.

4.1 Task A: Passage Retrieval

In the context of the information retrieval task, which follows a traditional ranked retrieval paradigm, the evaluation metric employed was the Mean Average Precision (MAP). Mean Average Precision (MAP) is a widely employed metric that is calculated across the entirety of a ranking(Voorhees, 2001). Instances where no answers are available were addressed by assigning complete credit to the system's "no answers" output and zero credit to all other responses. We have not trained the system since there is no method for fine-tuning the "text-embedding-ada-002" embedding model. With a threshold ranging between 0.4 and 0.95 with a 0.5 step, we found the best threshold to be 0.85 on the dev dataset with a 0.109438 MAP score and 0.267974 MRR score. The base model in this task is the BM25 model, which depends on the bag-of-words representation of the text (Amati, 2009). The BM25 model MAP and MRR scores for the dev dataset were 0.170291 and 0.313333 respectively. Using the test dataset, the BM25 model had a MAP score of 0.09036485 and an MRR score of 0.22603485 while our system

Task	Model	Score
Task A	BM25	0.090
	Similarity measurement with "ext-embedding-ada-002" embeddings	0.064
Task B	NWPR	0.326
	GPT-4-based Model	0.545

Table 1: Comparision between our methods and base models on the test dataset

achieved a MAP score of 0.06426543057 and an MRR score of 0.1608621226.

4.2 Task B: Reading Comprehension

The evaluation metric for Task B was the partial Average Precision (pAP) (Kishida, 2005), a rank-based measure designed to account for partial matching and assess the performance of a QA system in scenarios where the retrieved answer may not necessarily occupy the top rank and may only partially match one of the gold answers. Furthermore, pAP is well-suited for evaluating questions that may have one or more correct answers within the accompanying passage. This attribute makes pAP a more appropriate choice for assessing Task B compared to partial Reciprocal Rank (pRR) (Malhas and Elsayed, 2022). The baseline model to compare with is a naive whole passage retriever (NWPR) that returns the whole passage as an answer and has 0.255 and 0.3267900357 for the dev and test datasets respectively. Our GPT-4-based model scored better than the base model with pAP scores of 0.470 and 0.5393130538 for dev and test datasets respectively. Processing the results of GPT-4 gave a slice increase in performance when we tested it on the test dataset and got a pAP score of 0.5456830602. The GPT-3.5-based model yielded an exceedingly low score on the development dataset; consequently, we opted to exclude it from our comparative analysis.

Table 1 shows the results of our proposed methods compared to the corresponding base models.

5 Discussion

The results demonstrate that the OpenAI models utilized in this work provide a reasonable starting point for addressing the Qur'an QA tasks. However, there is substantial room for improvement to achieve state-of-the-art performance.

Regarding Task A, we initially attributed the low MAP score to a potential deficiency in Arabic language support. To investigate this, we employed Google Translate to render both the questions and passages into English. Subsequently, we applied the same methodology as described in Section 3.1. Surprisingly, the outcome proved to be notably inferior to the results obtained using the original Arabic dataset. We attribute this disparity to the inherent limitations of translation, which struggle to convey the precise nuances of Quranic passages accurately. Unfortunately, since "text-embedding-ada-002" embedding model is not open-sourced, it cannot be fine-tuned to fit our task.

In the context of the reading comprehension task, it is noteworthy that the GPT-3.5 prompt engineering approach performs notably worse than a naive baseline model. Conversely, the GPT-4 prompt engineering approach exhibits a significant performance improvement, surpassing the naive baseline by a considerable margin. However, it is essential to recognize that while GPT-4 demonstrates superior adherence to prompts compared to GPT-3.5, its behavior is not entirely deterministic, and variations can occur. Additionally, we must address the issue of "Prompt Injection", wherein a prompt could be introduced after the initial prompt, potentially altering the model's behavior. While this behavior was more prevalent in GPT-3.5, it is less pronounced in GPT-4. For instance, when applying the GPT-4-based model to the test set, we encountered very few cases like the question من هو المؤمن بتعريف القرآن ؟, which yielded the answer المؤمن بتعريف القرآن هو الذين آمنوا indicating that GPT-4 ignored entirely the prompt we mentioned in Section 3.2 and was appended before the question.

Conclusion

In this paper, we presented our methods for solving the two tasks of Qur'an QA 2023 Shared Task. We solved the passage retrieval task by (1) using "text-embedding-ada-002" embeddings to convert the questions and passages into a numerical representation, (2) calculating the cosine distances between the questions and answers, and then (3) selecting the top 10 similar passages. This method achieved a score lower than the baseline BM25 model with a MAP score equals to 0.06426543057. The reading comprehension task was solved using a handcrafted prompt along with GPT-4 with the *temperature* parameter equals to zero. Our method achieved a pAP score equals to 0.5456830602, approximately a 67% increase in performance compared to the baseline model.

Limitations

One of the limitations is the usage cost of ChatGpt APIs, especially GPT-4 which is approximately 10x the cost of using GPT-3.5. Another limitation is the explainability of the results. Providing explanations to answers is a challenging task and could be achieved partially by several methods as in (Zakieh and Alpkocak, 2021). However, the methods used in (Zakieh and Alpkocak, 2021) cannot be applied to the methods we used in this work.

References

Giambattista Amati. 2009. *BM25*, pages 257–260. Springer US, Boston, MA.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kenneth Ward Church. 2017. Word2vec. *Natural Language Engineering*, 23(1):155–162.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kazuaki Kishida. 2005. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan.

Rana Malhas and Tamer Elsayed. 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*, 19(6):1–21.

Rana Malhas and Tamer Elsayed. 2022. Arabic machine reading comprehension on the holy qur'an using cl-arabert. *Information Processing Management*, 59(6):103068.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2022. Qur'an QA 2022: Overview of the First Shared Task on Question Answering over the Holy Qur'an. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*, pages 79–87.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 Shared Task: Overview of Passage Retrieval and Reading Comprehension Tasks over the Holy Qur'an. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Rana R Malhas. 2023. *ARABIC QUESTION ANSWERING ON THE HOLY QUR'AN*. Ph.D. thesis.

OpenAI. 2023a. Embeddings - openai api. September 12, 2023.

OpenAI. 2023b. New and improved embedding model. September 12, 2023.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. 2012. Semantic cosine similarity. In *The 7th international student conference on advanced science and technology ICAST*, volume 4, page 1.

Maximilian Schreiner. 2023. Gpt-4 architecture, datasets, costs and more leaked. September 12, 2023.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational linguistics*, 37(2):351–383.

Marwan N. Swar. 2007. *Mushaf Al-Tafseel Al-Mawdoo'ee*. Dar Al-Fajr Al-Islami, Damascus.

GPT-4 Research Team. 2023. Gpt-4 technical report. Technical report.

Ellen M Voorhees. 2001. Evaluation by highly relevant documents. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82.

Abdul Razak Zakieh and Adil Alpkocak. 2021. Classification of medical transcriptions with explanations. Technical report, EasyChair.