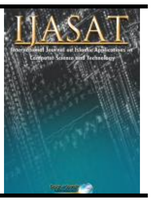


Using Automatic Question Generation Web Services Tools to Build a Quran Question-and-Answer Dataset

Sarah Alnefaie [1], Eric Atwell [2] and Mohammad Ammar Alsalka [3]

[1,2,3] University of Leeds, Leeds, UK
[2]King Abdulaziz University, Jeddah, Saudi Arabia
[1]scsaln@leeds.ac.uk, [2]e.s.atwell@leeds.ac.uk, [3]m.a.alsalka@leeds.ac.uk

Abstract

Question-and-answer datasets are essential in many fields, including questions related to the Quran, but there is still a lack of a Quran question-and-answer corpus. Therefore, this research paper aimed to create a valuable dataset for the research community using automatic question generation models. We first reviewed all the tools as black boxes, not as computational linguistics algorithms, compared them, and explored their features and drawbacks. We then identified freely available tools, which are the Explore AI Question Generation demo, the Cathoven Question Generator, the Questgen Question Generator, and the Lumos Learning Question Generator. Lastly, we created a corpus of Quran questions and answers using these web service tools. Our experiment indicates that these tools' performance varies in terms of many criteria, both the tools' performance in general and in terms of specific standards that measure the quality of the generated questions and answers. The Cathoven Question Generator was found to be the best tool in terms of general performance. Using these tools, we generated 40,585 questions and answers based on the English translation of the Quran.

Keywords: Natural language processing, Quranic natural language processing, automatic question generation, automatic assessment.

1. Introduction

People usually consult specialists to answer their questions. For example, people consult doctors to answer their medical questions, and so forth. In the same way, Muslims consult Islamic experts to answer religious questions. The answers include evidence from the Quran, considered a primary source in Islam. Muslims also turn to search engines to find answers to their questions, but due to the massive amount of information distributed over the internet, it takes a long time to visit all the relevant pages, which is a tedious process. Therefore, an Islamic question-answering (QA) system is required to provide quick and reliable answers to users' religious questions, rather than a list of webpages. The first step in the process of constructing such a QA system is to create a dataset.

According to Zhang et al. (2021), there is a relationship between a QA system's performance and the size and quality of its Question-and-Answer (Q&A) dataset. The larger the dataset, the higher the probability of finding an answer to a question, thus improving the system's performance. Therefore, the Quranic Q&A dataset should be large and of high quality.

There are several ways to create a Q&A dataset, such as using an automated question generation (AQG) algorithm. Automated question generation automatically generates a large number of high-quality questions and answers based on any passage of natural language text, thereby saving time, money, and effort compared to manually generating questions. Automated question generation models are available in the form of web service tools, mobile apps, or code. Many studies have used AQG models to create datasets, such as that of Fang et al. (2020), who suggested constructing open-domain Q&A pairs by using the AQG approach. In this paper, we analyse the effectiveness of using AQG tools to generate Q&A pairs for Quranic texts.

Based on the above, two research questions were formulated. The first question is as follows:

1. How can we create a large dataset of questions and answers on the Quran?

To answer this question, we set ourselves two objectives:

1.1 Review, evaluate, and compare the available AOG web services tools.

1.2 Use these tools to create a dataset of Q&A pairs for the Quran as a basic block for building a QA system in the future.

The second research question is as follows:

2. Do the AQG tools work with the Quranic text, considering its nature?

To answer this question, we set the following objective:

2.1 Evaluate the Q&A pairs that will be generated.

This paper is organized into several sections. Section 2 discusses collections of Quranic Q&A pairs, and section 3 presents the AQG approach and web services tools. Section 4 discusses the dataset for the experiment, while section 5 presents the experiment itself. The evaluation and results are described in section 6, while section 7 presents the discussion and analysis. The last section of this paper comprises the conclusion.

2. Background: Quranic Question-and-Answer Pair Corpus

Many studies have built a corpus of Q&A pairs based on the English translation of the Quran. Jilani (2013) assessed her QA system by constructing 100 Q&A pairs on the Quran. The source of this corpus was frequently asked questions on Islamic websites. The answers Jilani (2013) included in her dataset are not the original answers, as she deleted unnecessary verses and texts, based on her own judgment. One of the disadvantages of this dataset is that it is not publicly available.

Hamoud (2017) collected 1,500 Q&A pairs from Islamic websites, previous studies, and the fatwa service provided by the Mecca Grand Mosque. The data were then collected in one place, cleaned, and converted to a unified format. However, the number of questions is limited to 1,500 pairs and they are not available for use.

Adany (2017) gathered 263 Quranic questions and their answers by collecting questions from Islamic websites, as well as manually extracting questions by reading Quran texts. The Q&A pairs are in Arabic, with an added column that includes Yusuf Ali’s English translation of the answer verses. The limitations of this dataset are that it is small, unavailable, and covers only the chapters Al-Fatiha and Al-Baqarah.

Alqahtani (2019) built an Arabic corpus of 2,224 Q&A pairs on the domain of the holy Quran based on the book and websites to test and evaluate his QA system, known as the Annotated

Corpus of Arabic Al-Quran Question and Answer (AQQAC). The available version contains only 1,224 Q&A pairs. However, it covers 41% of the Quran with only one question, written in Arabic, and there are many questions using the format ‘Explain the following verse’.

Malhas and Elsayed (2020) constructed a gold standard Arabic corpus of 207 Q&A pairs based on the Quran called AyaTEC. To unify the QA system’s assessment process, they made it publicly available. They built AyaTEC by collecting questions from the internet and previous studies, with freelancers answering them, and the answers finally approved by religious scholars. Each answer refers to one or several verses. However, there are only 207 questions and 1,762 verses in Arabic.

Other studies, such as that of Ahmed et al. (2022) and Wasfey et al. (2022), created Quranic comprehension datasets where the record contains a question, a passage from the Quran, and an answer extracted from that passage, but these are in Arabic and not available.

The Quranic Reading Comprehension Dataset (QRCD) consists of 1,337 records, and was used in the ‘Quran QA 2022 Shared Task! Answering Questions on the Holy Quran’ to evaluate QA systems. It is available in Arabic (Malhas et al., 2022).

To the best of our knowledge, there is no available English corpus of Q&A pairs for the Quran. Therefore, the available AQG tools were used in this study to build such a corpus.

3. Questions Generator Web-Based Tools

Many web-based tools have been created to automatically generate questions and their short answers from any text passage. A comparison of these web service tools is presented in Table 1. The criteria used for the comparison are: (1) the size of the text allowed to be entered, (2) the ́question types that can be generated, (3) the number of questions that can be generated using the tool, (4) whether the tool is free or requires a subscription, (5) the language in which the text can be entered, (6) whether the tool is automatic or requires user intervention, (7) the time the tool requires to complete its task based on our experiment, and (8) the algorithm used by this tool to generate the question. These tools are discussed below.

The Joint Information Systems Committee (JISC) developed a question generator tool called the ExploreAI Question Generation (QG) demo.1 This demo allows users to enter a paragraph limited to 1,000 characters and uses the Text-To-Text Transfer Transformer (T5) to automatically determine the answers from the text and generate the right questions for these answers. The generated questions are then displayed. In addition, it allows the user to answer questions and then display the correct answers or display the correct answers directly. Between one and four questions are generated for each paragraph, and the question generation process takes 75 seconds. One disadvantage of this system is that any grammatical or factual errors in the text will appear in both the answer and the question. This code is available.2

Cathoven’s team built a question creator tool called the Cathoven Question Generator (QG).3 Users have to enter the text and the number of questions they want to generate. This tool can accommodate a text passage of up to 500 words as input. There are three options available with regard to the number of questions to be generated, namely a ‘ few ’, ‘ many ’, and ‘ tonnes ’. Then, using the text-to-text generation natural language processing (NLP) model, the encoder-

[1] https://exploreai.jisc.ac.uk/tool/question-generation
[2] https://github.com/patil-suraj/question_generation/
[3] https://hub.cathoven.com/?scene=generator

decoder structure is applied to analyse the details of the text and quickly generate the question. It usually takes a minute to produce questions, and then the questions only are displayed. When the option 'many' is chosen, between one and 20 questions will be generated. The user can answer the questions, and the model will indicate whether the answer is correct or incorrect. This model is smart enough to consider the user's answer valid if it has the same meaning as the correct answer. The user can also view the correct answers. The questions and answers can be exported as TXT files.

The Questgen Question Generator (QG) tool4 is built on a combination of computational linguistics and advanced AI algorithms. Many transformer models are used, such as GPT-3, GPT-2, T5, and BERT. It is available as an application, an API, and from an open-source NLP library. There is a free trial application that provides only 15 free tries. To have access to unlimited tries and features, a subscription to the application and the API is required. The Questgen QG open-source library is less accurate than the application and the API. Three types of questions can be generated, namely multiple-choice questions (MCQs), Boolean questions (yes/no), and short-answer questions. The user needs to insert a text of 50 to 500 words and select the type of questions, after which the questions and answers will appear within seven seconds.

Lumos Learning is an Edtech platform that attempts to improve the learning process using digital solutions. One of its most important features is the ability to generate assessments for reading comprehension, mathematics, and language and grammar using the Lumos Learning Question Generator (QG) Tool.5 After a passage of at least 2,000 characters has been pasted into the tool, the machine learning algorithms generate, on average, between zero and 32 questions with their answers. The generated assessment can be exported in a CSV format. The other tools listed in Table 1 require a subscription.

Table 1: Comparison of web services question generation tools.

Tools	Input Size	Types of Questions	Number of Questions	Free	Language	Automatic Tool	Duration	Algorithms
ExploreAI Question Generation demo	1,000 characters	Short answer	1 to 4	Yes	English	Yes	75 seconds	Text-To-Text Transfer Transformer (T5)
Cathoven QG	500 words	Short answer	1 to 20	Yes	English	Yes	60 seconds	Text-To-Text generation model
Questgen QG	50 to 500 words	MCQ yes/no, short answer	0 to 18	Application and API: no Open source NLP library: yes	English	Yes	7 seconds	GPT-3, GPT-2, T5, and BERT

[4] https://questgen.ai/

[5] https://www.lumoslearning.com/llwp/admin/kb_qa_generator.html?sid=337516&TP_ID=50497835_0_catid5069s

Lumos learning QG	At least 2,000 characters	Short answer	0 - 32	Yes	English	Yes	40 seconds	-
QuestionAid QG [6]	1,500 characters	Short answer	5	No	24 languages	Yes	38 seconds	-
PrepAI QG [7]	No limit	MCQ Fill the blank Short answer True/false	Easy MCQ 36 Medium MCQ 51 Hard MCQ 2 Fill in the blank 18 Short answer 0 True/false 36	No	English	Yes	1 minute and 29 seconds	-
Quillionz QG[8]	300-3,000 words	MCQ Fill the blank Short answer	MCQ 26 Fill in the blank 18 Short answer 10	No	English	No need for user intervention to determine possible keywords	1 minute and 54 seconds	-
Automatic QG[9]	-	MCQ Descriptive True/false	MCQ 31 Descriptive 2 True/false 2	no	English	No need for user intervention to determine possible keywords	-	-

4. Dataset

This section provides an overview of the dataset used in this paper. This experiment was conducted on two English versions of the Quran: Yusuf Ali and Sarwar, which were taken from Tanzil[10] as text files containing 6,236 verses. Each verse is displayed on a single line. We divided the verses into groups, and each group was placed on a separate line as an input to the tool, because some tools have minimum or maximum requirements for the entered text.

5. Experiment

The following methodology was followed:

1. Identify the free and available web services tools from among the following tools: The Explore AI Question Generation demo, the Cathoven QG, the Lumos Learning QG, and the Questgen's open-source library. However, these tools do not consider Quranic text specificity, such as the existence of similar verses. Therefore, they may not work appropriately for the Quran.

2. Divide the verses into groups to fit the minimum and maximum limits of text that can be entered into the tool each time. The Explore AI Question Generation demo requires the input text size not to exceed 1,000 characters. At the same time, the number of words should not exceed 500 in the Cathoven QG. The text should range between 50 and 500 words for the

[6] https://www.question-aid.com/collections/205

[7] https://app.prepai.in/generate-questions

[8] https://app.quillionz.com,

[9] https://softwarecountry.com/our-products/automatic-question-generator/

[10] https://tanzil.net/

Questgen QG. The least number of words that can be entered into the Lumos Learning QG is 2,000.

3. Enter the text into the tools.

4. Export the generated questions and answers to Excel files.

6. Evaluation and Results

This section defines the criteria we used for the evaluation process and the results of this process. We used two types of standards: The first type measures the quality of the questions and answers, while the second type measures the performance of the tools in general.

6.1 Measuring the Quality of the Generated Questions and Answers

6.1.1 Automatic Evaluation

First, we evaluated the quality of the generated questions and answers using the available datasets. There are three available datasets, namely AyaTEC (Malhas and Elsayed, 2020), QRCD (Malhas et al., 2022), and AQQAC (Alqahtani, 2019). These datasets are in Arabic, so we could not directly use them to assess AQG systems that work with English only.

6.1.2 Human Evaluation

According to Amidei et al. (2018) and Zhang et al. (2021), the most popular human evaluation methodology is the eliciting expert judgement method. We applied this methodology by asking three annotators to rate 200 questions randomly selected from the generated questions, based on a number of criteria, using a rating scale from 1 to 3, where 1 is the worst grade and 3 is the best. Next, we calculated an average score for each question and then an average score for all questions. For example, to calculate the score of the syntactic correctness standard for the Cathoven QG tool, each annotator read the first question generated by this tool and gave a score between 1 and 3 for the syntactic correctness of the question. Then we added up the scores of the three annotators for the first question and divided it by their number, which is three, to get the average score for the first question. Finally, we added all 200 average scores of the questions and divided the total by 200 to get the average score for the standard of syntactic correctness for the Cathoven QG tool.

The evaluation criteria we used to measure the quality of the generated questions and answers are syntactic correctness, semantic correctness, ambiguity, relevance, difficulty, and answerability. Syntactic correctness means the generated questions and answers are grammatically correct. At the same time, semantic correctness means that the question and the answer are meaningful. The question's ambiguity also has to be assessed if it is asked independently without the text, because the lack of ambiguity of the question leads to a clear answer. The relevance of the answer and the question to the context is also significant. The difficulty was measured to ensure that the reader fully understands the entered text and needs some logic to answer the question. In addition, an answerability criterion was established to verify that each answer is a possible answer to the created question.

The quality results of the questions and answers generated by the four tools are listed in Table 2. The Cathoven QG tool was dominant in all criteria except the relevance, where it was outperformed by the ExploreAI QG demo by a narrow margin of 0.2%. The ExploreAI QG demo achieved the second-highest score in most criteria, except the difficulty criterion. It was beaten by the Lumos Learning QG, which received the third-highest score in the rest of the standards. The Questgen QG results are at the bottom of the list.

Table 2: Human evaluation results for questions and answers generated based on quality standards.

Criteria	ExploreAI Question Generation demo	Cathoven QG	Lumos Learning QG	Questgen QG
Syntactic Correctness	2.43	2.70	2.42	2.30
Semantic Correctness	2.22	2.44	2	1.99
Ambiguity	2.26	2.52	2.15	2.14
Relevance	2.98	2.96	2.86	2.84
Difficulty	1.08	1.20	1.16	1.07
Answerability	2.41	2.51	2.09	2.22

We used the Fleiss kappa to measure the agreement between annotators (Landis and Koch, 1977; Sim and Wright, 2005). The agreement value between the annotators was 0.786, which is considered a substantial agreement.

6.2 Measuring the Performance of Tools

We also measured the tools' performance in general, using several criteria, including the variety of question types, usability, the total number of questions, and the actual duration if we assume that the tools are running concurrently. First, the types of generated questions vary, one of the indications of a tool's power. Usability means how easily the user can use a tool effectively and efficiently. The total number of questions generated for the entire Quran is also an important criterion, because it is an indicator of performance. The last metric refers to the time it takes to create questions for the whole Quran.

Table 3 presents the results of the performance of the four tools based on the experiments conducted to generate Q&A for the Holy Quran. The top tools are the Cathoven QG and the Lumos Learning QG, based on two measures. The Cathoven QG is the best tool in terms of the diversity of question types and ease of use. By comparison, the Lumos Learning QG is the best tool in terms of the total number of questions generated and duration. It was followed by the ExploreAI QG demo and the Questgen QG, which are better in terms of other scales. The ExploreAI QG demo is the best in terms of its ease of use, while the Questgen QG is the best in terms of duration. On average, the Lumos Learning QG and the Cathoven QG outperformed the ExploreAI QG demo and the Questgen QG. The difference in superiority is only in terms of one criterion.

Table 3: Human evaluation results for the tools' performance.

Criteria	ExploreAI Question Generation demo	Cathoven QG	Lumos learning QG	Questgen QG
Variety of Question Types	Who, What, How, Where, Whose, Why, When, Which, Whoever	**Who, What, How, Where, Whose, Why, When, Which, Whoever, Whom**	Who, What, How, Where, Whose, Why, When, Which, Whoever	Who, What, How, Where, Whose, Why, When

Usability	**It is easy to use**	**It is easy to use**	The user needs time to learn how to use the tool.	A user with a background in Python programming can handle the tool and complete the task successfully and efficiently.
Total Number of Questions	Sarwar: 2,761 Yusuf Ali: 4,174 Total: 6,935	Sarwar: 6,815 Yusuf Ali: 5,902 Total: 12,717	**Sarwar: 8,537** **Yusuf Ali: 5,062** **Total: 13,599**	Sarwar ; 3,675 Yusuf Ali: 3,659 Total: 7,334
Duration	Five days	Two days	**12 hours**	**12 hours**

To summarize the above, on average, the Cathoven QG is the best in terms of tool performance and the quality of the generated text. Regarding the other tools, the ExploreAI QG demo and the Lumos Learning QG are at the same level, and then the Questgen QG.

7. Discussion and Analysis

This section analyses the results obtained by the four web services tools and discusses the possible reasons for these results.

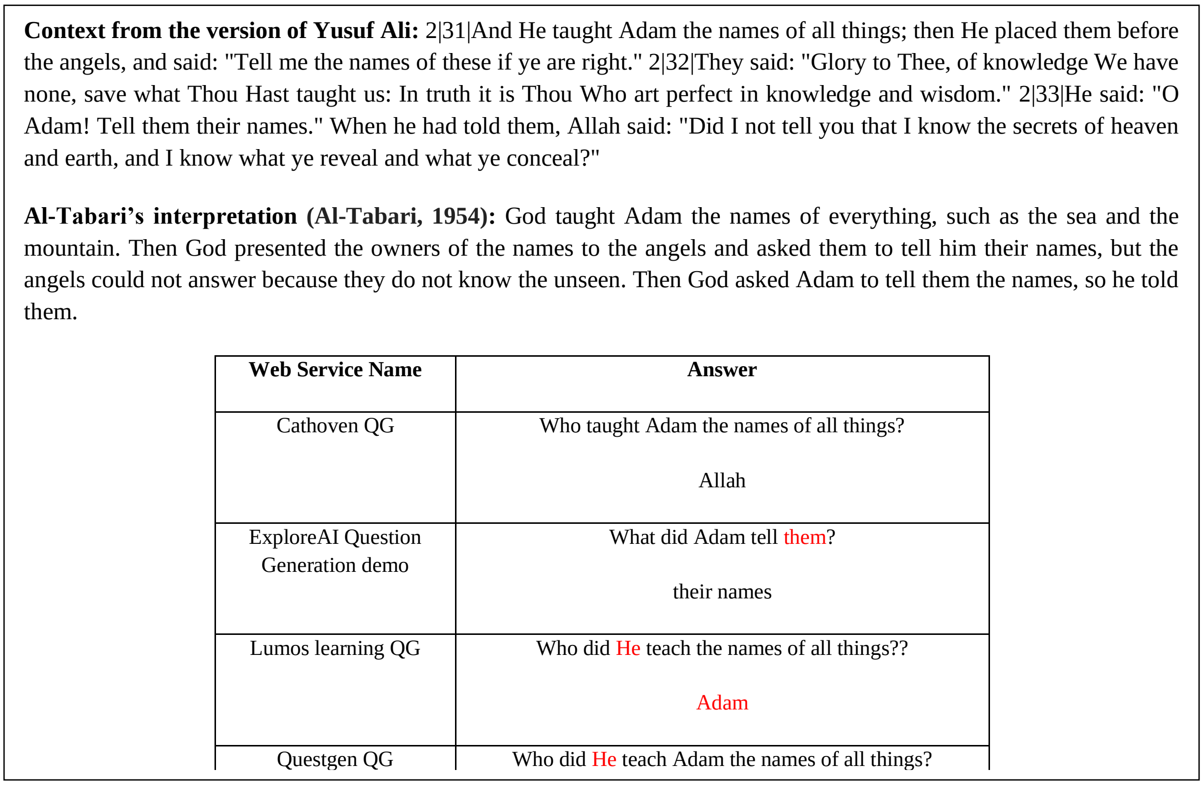
As previously stated, the Cathoven QG obtained the highest results compared to the rest of the tools in terms of the quality of the generated text. For example, when we entered a number of verses, we found that each tool generated a different Q&A, as shown in Figure 1. The Cathoven QG created a Q&A that is correct in terms of meaning and syntax, complete in terms of relevance to the text, and clear in terms of understanding; there is no ambiguity when asking the question independently. By comparison, the other tools suffered from problems: First, there was incorrect sentence structure or meaning, for example, with the Lumos Learning QG and the Questgen QG. Second, the ambiguity was sometimes due to the presence of pronouns in the question or answer, such as with the ExploreAI QG demo. Finally, sometimes there was a wrong answer, such as with the Lumos Learning QG and the Questgen QG.

All the tools generate texts that suffer from issues affecting their performance. The difference between them is the type of problems and the amount of text that gives rise to issues, such as these examples:

- Sometimes the Cathoven QG generates incomplete answers. For example, it generated the following question and answer from the script below:
  Question: ‘What did Satan do to the children of Israel?’
  Answer: ‘make them slip from the (garden) and get them out of the’
  Script: ‘Then did Satan make them slip from the (garden), and get them out of the state (of felicity) in which they had been. We said: “Get ye down, all (ye people), with enmity between yourselves. On earth will be your dwelling-place and your means of livelihood – for a time.”’ (Verses 36 of Surat Al-Baqarah in the version of Yusuf Ali)
  However, the **complete, correct answer** is ‘make them slip from the (garden) and get them out of the state (of felicity)’.

- One of the problems we encountered were incorrect answers. For example, the ExploreAI QG demo tool generated the following Q&A:
  Question: ‘Who is an enemy to Allah and His angels and messengers?’

Answer: ‘Gabriel and Michael.’
Script: ‘Whoever is an enemy to Allah and His angels and messengers, to Gabriel and Michael, − Lo! Allah is an enemy to those who reject Faith.’ (Verse 98 of Surat Al-Baqarah in the version of Yusuf Ali).



Context from the version of Yusuf Ali: 2|31|And He taught Adam the names of all things; then He placed them before the angels, and said: "Tell me the names of these if ye are right." 2|32|They said: "Glory to Thee, of knowledge We have none, save what Thou Hast taught us: In truth it is Thou Who art perfect in knowledge and wisdom." 2|33|He said: "O Adam! Tell them their names." When he had told them, Allah said: "Did I not tell you that I know the secrets of heaven and earth, and I know what ye reveal and what ye conceal?"

Al-Tabari’s interpretation (Al-Tabari, 1954): God taught Adam the names of everything, such as the sea and the mountain. Then God presented the owners of the names to the angels and asked them to tell him their names, but the angels could not answer because they do not know the unseen. Then God asked Adam to tell them the names, so he told them.

Web Service Name	Answer
Cathoven QG	Who taught Adam the names of all things? Allah
ExploreAI Question Generation demo	What did Adam tell them? their names
Lumos learning QG	Who did He teach the names of all things?? Adam
Questgen QG	Who did He teach Adam the names of all things?

Figure 1. A sample of generated questions using the Cathoven QG, the ExploreAI QG demo, the Lumos Learning QG, and the Questgen QG for the same verses.

As we can see, the answer is wrong: The correct answer is ‘those who reject Faith’. The reason for this incorrect answer may be that this tool cannot understand and analyse bracketing commas.

- The Lumos Learning QG generated the following Q&A:
  Question: ‘Who will not grasp the Message?’
  Answer: ‘Men of understanding.’
  Script: ‘He granteth wisdom to whom He pleaseth; and he to whom wisdom is granted receiveth indeed a benefit overflowing; but none will grasp the Message but men of understanding.’ (Verse 269 of Surat Al-Baqarah in the version of Yusuf Ali)
  As we can see, the question is wrong, and **the correct question** is: ‘Who will grasp the Message?’ We have to conclude that Lumos Learning QG could not correctly analyse the negation and affirmation texts.

- Sometimes the answer appears in the question, and the grammatical structure of the question is incorrect, as in the following Q&A generated by the Questgen QG:
  Question: ‘Who did Adam learn from his Lord’s words of inspiration?’
  Answer: ‘Adam.’

Script: 'Then learnt Adam from his Lord words of inspiration, and his Lord turned towards him; for He is Oft-Returning, Most Merciful.' (Verse 37 of Surat Al-Baqarah in the version of Yusuf Ali)

Regarding the results of the four web services and their performance in general, as presented in Table 3, we can note several points:

- We noticed that all the tools generated the same types of questions: Who, what, how, where, whose, why, and when, while the ExploreAI QG demo and the Lumos Learning QG tools can also generate questions that start with whoever and which. In addition to all of the above, the Cathoven QG tool can generate questions that begin with whom.
- The Cathoven QG and the ExploreAI QG demo were easy to use. The user only needs to copy-paste text and click the button, but the Lumos Learning QG requires registration and time to get to the tool's location. The Questgen QG requires the user to have programming skills in Python to use the library.
- The Cathoven QG and the Lumos Learning QG tool generated a similar number of questions, around 13,000, while the Questgen QG and the ExploreAI QG demo produced half that number.
- The Lumos Learning QG and the Questgen QG were the fastest tools to run.

The main drawback of these tools is that they generate an enormous number of questions and answers containing many different errors that need to be manually reviewed and revised.

8. Conclusion

This paper reviewed web services tools that automatically generate questions from a text. We used the available tools to create a corpus of questions and answers for the Quran in English. The experiment was conducted using four tools, namely the ExploreAI QG demo, the Cathoven QG, the Lumos Learning QG, and the Questgen QG. The results indicated that the Cathoven QG, the ExploreAI QG demo and the Lumos Learning QG perform at the same level, while the Questgen QG was slightly less effective, based on the tools' performance in general and the quality criteria for the generated questions and answers. The Cathoven QG outperformed the other tools based on the quality of the generated questions and answers, with an average score of 2.70 for syntactic correctness, 2.44 for semantic correctness, 2.52 for ambiguity, 1.20 for difficulty, and 2.51 for answerability. In addition, it ranked first based on the tool's overall performance. The four tools combined generated 21,788 questions and answers for the Sarwar Quran and 18,797 for Yusuf Ali.

References

Adany, M. A. (2017). An automatic question answering system for the Arabic Quran. Ph.D. thesis, Sudan University of Science and Technology.

Ahmed, B. H., Saad, M. K., and Refaee, E. A. (2022). QQATeam at Qurán QA 2022: Fine-Tunning Arabic QA Models for Qurán QA Task. Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022).

Alqahtani, M. M. (2019). Quranic Arabic Semantic Search Model Based on Ontology of Concepts. Ph.D. thesis, University of Leeds.

Al-Tabari, M. B. J. (1954). جامع البيان عن تأويل القرأن. Dar Al-Fikr.

Amidei, J., Piwek, P., and Willis, A. (2018). Evaluation methodologies in automatic question generation 2013−2018.

Fang, Y., Wang, S., Gan, Z., Sun, S., Liu, J., and Zhu, C. (2020). Accelerating real-time question answering via question generation. ArXiv Preprint ArXiv:2009.05167.

Hamoud, B. I. (2017). A Question Answering System Design about the Holy Quran. Ph.D. thesis, Sudan University of Science and Technology.

Jilani, A. (2013). Parallel Corpus Multi Stream Question Answering with Applications to the Qu'ran. Ph.D. thesis, University of Huddersfield.

Landis, J. R., and Koch, G. G. (1977). The measurement of observer agreement for categorical data. Biometrics, 159−174.

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, Jun). Qur'an QA 2022: Overview of the first shared task on question answering over the holy qur'an. Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022).

Sim, J., and Wright, C. C. (2005). The kappa statistic in reliability studies: use, interpretation, and sample size requirements. Physical Therapy, 85(3), 257−268.

Wasfey, A., Elrefai, E., Marwa, M., and Haq, N. (2022). Stars at Qur'an QA 2022: Building Automatic Extractive Question Answering Systems for the Holy Qur'an with Transformer Models and Releasing a New Dataset. Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022).

Zhang, R., Guo, J., Chen, L., Fan, Y., and Cheng, X. (2021). A Review on Question Generation from Natural Language Text. ACM Transactions on Information Systems (TOIS), 40(1), 1–43.