EasyChair Preprint

№ 8588

Arabic Automatic Question Generation Using Transformer Model

Saleh Alhashedi, Norhaida Mohd Suaib and Aryati Bakri

August 3, 2022

Arabic Automatic Question Generation Using Transformer Model

Saleh Alhashedi[1][0000-1111-2222-3333], Norhaida Mohd Suaib[2][1111-2222-3333-4444] and Aryati Bakri[3][1111-2222-3333-4444]

School of Computing, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
[2]UTM Big Data Center, Ibnu Sina Institute of Scientific and Industrial Research (ISI-SIR),
[3]Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
[1]salehhashedi@gmail.com, [2]haida@utm.my and [3]aryati@utm.my

Abstract. Questions play a vital role in the educational assessment process and enhance learning outcomes for students of all ages. Preparing question exams is a challenging and time-consuming task that requires a thorough comprehension of the topic and the ability to construct the questions, which becomes more difficult as the text size increases. Automatic question generation (AQG) is the task of generating natural relevant questions from diverse text data inputs with optionally giving an answer. A few contributions have been made to address this issue in Arabic language. Many previous works rely on manually constructing question styles using Rule-based methods and input text from kids' books, stories, or textbooks. These models are limited in linguistic diversity, and the tasks become more complex and challenging when the text size becomes more extensive. Transformer is one of the most adaptable deep-learning models, having been successfully applied to various Natural Language Processing (NLP) tasks. In this paper, we proposed an end-to-end Arabic automatic question generation (AAQG) model based on the Transformer architecture to generate N interrogative questions for educational content from a single unlimited-length document.

Keywords: Arabic Question Generation, Transformer, NLP, TextRank, Sentence Extraction.

1 Introduction

The traditional educational methods have evolved over the previous few decades. Many educational institutions and organizations have shifted to online education as a mode of operation. They provide examinations after each lesson to ensure that students attend and grasp the material. Furthermore, online schools and colleges update the educational curriculum at each stage to maintain the changes and elevate the educational system's level to keep up with new technologies. In addition, smart curriculums and Open Online Courses (MOOCs) are limitless resources, and the spread of Arabic educational content on the internet is increasing daily.

Questions play a significant role in educational operations and help to improve learning results for students of all ages. People realized that giving students tests helps

improve learning outcomes and gauge students' knowledge. Therefore, the online instructional content undergoes revisions and upgrades, either by introducing new resources or upgrading old ones. On the other hand, manually constructing question tests is a complex undertaking for many examiners. It is a time-consuming task that requires skill, experience, and comprehensive acquaintance with the subjects. Under the circumstances of the educational process's adaptation and the rapid growth of the Arabic educational content on the internet, and to take advantage of these materials. The old method impedes institutions and academic organizations; we need to automatically prepare questions tests and make the questions preparation process much more manageable by developing a simple and effective solution to accomplish this task and save time and effort [1].

Question generation (QG) is a critical and continuing task in natural language processing (NLP) [2, 3]. It is a task of creating one or more sorts of questions: interrogative questions, correctness and incorrectness questions, free-response questions, multiple-choice questions, and fill-in-the-blank questions, by utilizing an input text with optionally provided an answer [3]. This task is more established in English than Arabic because the Arab community research has made fewer contributions to using Artificial Intelligence (AI) to address various difficulties that have been solved in other international communities, including QG tasks. According to the review by Kurdi et al. [4] review, 51 studies in English, five in Chinese, and just a few works in Arabic have been accomplished [5-7].

The earlier works in Arabic question generation relied on the Rule-based method, which usually depends on the manually designed and constructed process to generate questions. While Elbasyouni et al. [7] cleaned Arabic complex text morphology by diacritic and normalizing Arabic, they used expressions with regular expressions and Gazetteer Arabic Named Entity for generation task. Bousmaha et al. [6] Used kids' stories and combined the semantic role labeling with question-based models to capture the sense of sentences for Arabic language grammar. Alazani & Mahender [5] used a Rule-based approach of Part-of-speech (POS) tagger and Named Entity Recognition (NER) to generate linguistic questions using text from the fifth-grade textbook.

The automatic question generation (AQG) models in the previous and current works suffer from poor performance and low-quality question generation outcomes due to using complex models and manually constructing templates for the generation task. They need many pipelines and features to optimize the model when the text size becomes more extensive. To add more, those models are constrained in terms of their linguistic diversity. They need massive data annotation for the training process, which is time-consuming and requires domain-specific knowledge and experience.

In numerous deep learning and natural language processing (NLP) tasks, including QG, Transformer-based models, have exceeded all prior neural network models. Those models can learn to predict a relevant question by themselves throw the training process without manually constructing questions or using question templates. The first sequence transduction model relies entirely on attention [8]. They are fine-tuned, and pre-training models like BERT Devlin et al. [9] have shown incredible accuracy and better performance on smaller datasets in different types of NLP tasks in other languages.

The earlier and current works to generate questions in Arabic language were built and tested using limited input texts, from kids' stories [6] and textbooks of elementary school [5] or a sentence from a paragraph [7]. The necessity to provide an ultimate solution to this issue is to generate an N number of questions using any text length. TextRank is a graph-based ranking algorithm for a single unlimited text size document to extract keyword and sentence for automatic text summarization by selecting the most important sentences from the given text. It showed better results and outperformed other statistical approaches using TF-IDF [10].

2 Related Works

2.1 Question Generation

A rule-based approach was and is still used in early QG works. It relies on human design to extract features from the input text. They are good at capturing linguistic relationships between words to explain the sentence. Learning from data requires extensive data annotation, which is time-consuming, requires domain-specific knowledge and experience, and is constrained in terms of linguistic diversity.

In the recent growth of deep-learning, neural networks and NLP tasks, question generation approaches have gained much more attention in NLP tasks. They have shown great success, accuracy, and performance on a small dataset, including AQG. Transformer models learn how to predict the question using the datasets and an optionally given answer. To add more, unlike traditional approaches, which must be learned from scratch, many neural network-based methods such as BERT [9], GPT2 [11] and T5 [12] are unsupervised pre-trained and fine-tuned on source tasks as a starting point. Those models applied the learned knowledge to the target tasks to save time and effort. They achieved remarkable success in many NLP tasks.

For question generation tasks, Varanasi et al. [13] proposed a copy mechanism for BERT-base model with a shared encoder-decoder for question generation tasks. They used paragraph, answer and question as input sequence tokens. A semi-diagonal mask is utilized for information flow control. A copy-mechanism was utilized over the input text to obtain attention probabilities as normal copy. Then a self-copy was used to weight all self-attention average and the generation probability combined by copy-generate probability. The proposed model was trained on SQuAD dataset and achieved 22.7 score in BLUE4 metric.

Liu & Zhang. [14] introduced a Chinese sentence fill-in-the-blank question generation dataset by employing three baseline models based on LSTMs and BERT for the generation task, they compressed the long Chinese text and broke the passage into sentences that contain the answer, and they achieved a state-of-the-art for Chinese language with 30.15 score in BLUE2 and 12.18 score in BLEU4 metrics and the model trained on WebQA dataset. Xiao et al. [15] proposed a state-of-the-art ERNIE-GEN framework to enhance multi-flow sequence to sequence pre-training and fine-tuning framework by using only a single pre-trained language model to tackle the exposure bias problem in current pre-training tasks, including question generation in English

Language. The model was trained in SQuAD dataset and achieved 25.41 score in BLUE4 metric. Fatih et al. [16] used a single fine-tuned text-to-text Transformer multilingual T5 for automatic QA and QG using the Turkish QA dataset without label answers. The proposed model achieved a state-of-the-art for Turkish language with 32.8 score in BLEU2 metric.

Generate questions from text only using neural networks and due to the real-life where the contents are taken or crawled from websites without answers. Most generation tasks concentrate on relying on the answer as a guide to identify a specific fragment of the input text that includes the information [14, 17-19].

Other question generation works from content texts show improvement in the performance. Wang et al. [20] proposed a framework for question generation with worthy phrases input only. They generate questions by extracting multiple phrases using a rule-based model that relies on POS and applying Pointer Network to select input sentences using bi-directional LSTM as an encoder and another LSTM to decode the phrases. They used a message-passing module to map the extracted phrases to the generation agent to generate the questions. Ghanem et al. [21] proposed a model for generating questions without answers and only used the stories to train the T5 model on the SQuAD and CosmosQA datasets. To control the style of the questions, they use a rule-based extractor to sample the given text meaningfully.

2.2 Text Extraction

Text extraction is a process of summarizing the long text by extracting the most important sentences from the input text. Generating relevant questions from long text or a single document becomes challenging for Transformers models such as BERT, which can take up to 512 embedded tokens. To tackle this issue, Zhao et al. [17] trained BERT-base model to generate event-centric summarization from an educational paragraph by adding two control signals, one for the question type and the other one for the question order before the input paragraph.

Gollapalli et al. [22] created PaintTeR algorithm scores to compute the text span by random walks throw a representation probability moving matrix on a long document to compute the move from one word to another. Then, they combined the associated words with the painting document and ranked the passage spans using a word co-occurrence graph. Wijanarko et al. [23] proposed three steps to extract key-phrase using Naïve Bayed model. First, they cleaned the document and stemmed the phrases after breaking down the document into phrases, and then, they used TF-IDF algorithm score to measure the specific phrase in the input document. Finally, they compute the distance of the phrases base on the number of words present in the first appearance and divided by the number of document words.

3 The Proposed Model

This academic work attempts to extend the previous work to generate questions automatically from long Arabic texts using Transformer BERT-base model without a

template or Human-crafted rules by only using free-size text as input to generate N number of questions.

The proposed model is classified into two main categories, as illustrated in Figure 1. The first part is the question generation model, which is used as a base model for the Arabic automatic question generation task. The second part is to solve the base model limitation and to generate an N number of questions from a long text by extracting the key sentences from the long text as input to the core model.

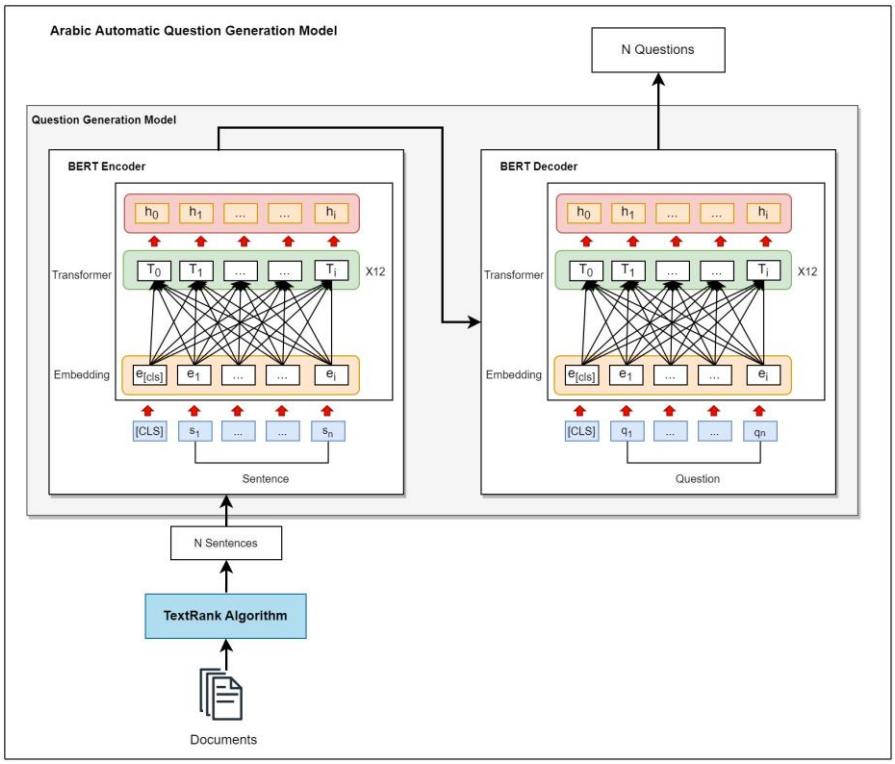

Fig. 1. The proposed model architecture.

3.1 Arabic Automatic Question Generation Model

BERT is a language representation model from Transformer, which stands for Bidirectional Encoder Representations. It has been pre-trained on tremendous amounts of data, with BERT-base trained on 800+ million words from BooksCorpus and BERT-large trained on 2500+ million words from English Wikipedia. It understands how to represent text and can process a large amount of text and language. The model can scan left and right many times before producing a vector representation for the context of a word, and it can return different vectors for the same word depending on the words around it. Providing self-attentions encode syntactic information and can be probed for dependency parsing and POS tagging, and spaCy's Named Entity Recognition locates and

categorizes named entities in unstructured text. To add more, it learns different linguistic features at different layers. It outperforms legacy methods and has the potential to improve performance and reduce training costs by sharing the weights.

Motivated by the Transformers BERT model's recent success in many NLP tasks. We re-implemented Rothe et al. [24] shared BERT-base encoder-decoder architecture as a base model, as shown in Figure 2. BERT-Shared is an encoder-decoder based on BERT's architecture. While the encoder is the same as BERT's architecture, the decoder has some changes. In each BERT block, the cross-attention layers are randomly initialized and added in between the self-attention layer and the feed-forward layers. To comply with auto-regressive generation, The bi-directional self-attention layers were changed to uni-directional self-attention layers. On top of the last block of the decoder, an LM Head layer was added with typically the same weight of word embedding.

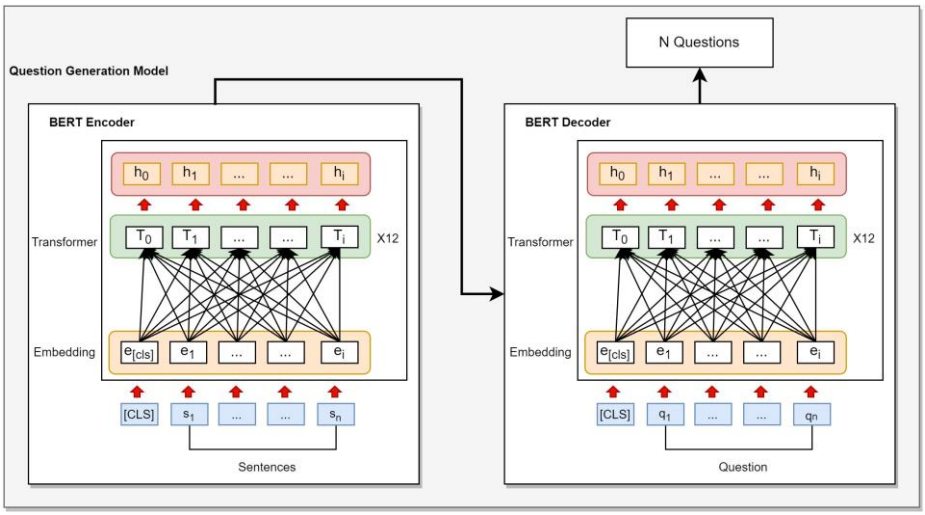

Fig. 2. Arabic question generation base model: we re-implemented Rothe et al. [24] shared BERT-base encoder-decoder architecture, adapted the model, and tokenized the text using Arabert Antoun et al. [25] to support Arabic language.

The base model encoder and decoder were adapted and initialized with Arabert Antoun et al. [25] checkpoint to support Arabic language. AraBERT model was used for language understanding and trained on 70 million sentences containing 3 billion words from 23GB of Arabic text. We tokenized the text to support Arabic language and adapted the model for the question generation task with input text and without an answer. In addition, we initialized all weights from AraBERT checkpoint, except the encoder-decoder attention variable, which was initialized randomly.

To better understand the question generation task, let S donate to the sentence and Q donate to the question targeting the sentence. The problem formulation is:

$$Q = P(Q|S) \tag{1}$$

3.2 Generate N Number of Questions from a Single Unlimited-Length Document

TextRank is a graph-based ranking algorithm for a single document to extract keyword and sentence for automatic text summarization. The sentences are represented as nodes by an undirected connected graph. The recommended sentence is calculated by the number of words that are common in two sentences using the TextRank algorithm [26]. They are sorted based on their scores, and the highest-ranked sentences will be selected [27]. This work employed TaxtRank algorithm to extract the important sentences from each paragraph in a single long document to generate N number of questions and to tackle the limitation of BERT model, which can take up to 512 tokens (words), as shown in Figure.3.

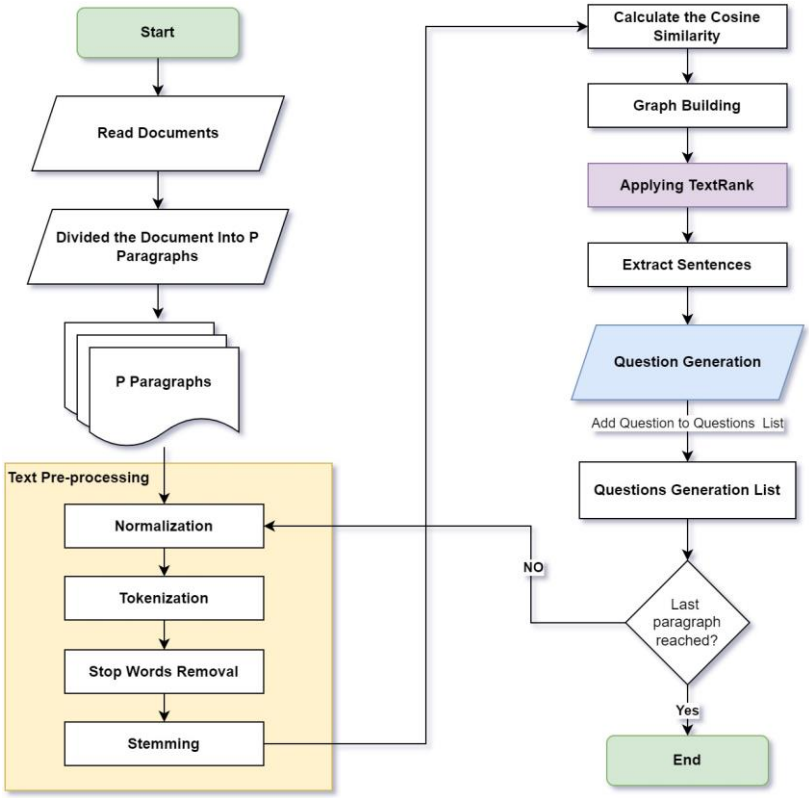

Fig. 3. Apply TextRank algorithm.

Apply TextRank Algorithm. The method starts by reading the document and dividing it into paragraphs. Then, TextRank algorithm was employed to extract the key sentences from each paragraph. After that, all the extracted sentences list are donated to the base model to generate the questions and stored in a list until the generation processes finish for the whole list. This task contains six stages:

- **Reading the Documents:** This stage starts by reading the documents.
- **Dividing the document into paragraphs**
- **Pre-processing contains the following process:**
  - Text Cleaning: remove Arabic diacritics, Tashkeel, Tatweel from the sentences.
  - Normalization: remove all no Arabic alphabet letters, digits and punctuation from the sentences.
  - Tokenization: split the string into individual words without blanks or tabs.
  - Remove stop words: remove stop words and punctuation from the string to reduce the text.
  - Stemming: converting a word to its most general form or stem.
- **Calculate the cosine similarity using equation 2.**

$$Similarity\ (S_i, S_j) = \frac{\left|\{w_k | \, w_k \in S_i \,\&\, w_k \in S_j\}\right|}{log(|S_i|) + log\ (|S_j|)} \quad (2)$$

The ranking model recommends a sentence by calculating the number of words N_i in a sentence $S_i = w_1^i, w_2^i, \ldots \ldots w_n^i$ that are common in two sentences S_i and S_j. They are sorted based on their score, and the sentences with the highest ranked are the selected sentences.

- **Build the graph.**
- **Apply TextRank Algorithm:** the algorithm extracts the key sentences from the paragraphs and store them in a list. They are sorted depending on their score, and the highest-ranked sentences are selected once.

4 Experiments

4.1 Setup and Fine-Tunning

All fine-tune processes and experiments for the proposed model have been performed using NVIDIA RTX A6000 GPUs and 48GB with RAM using Transformers Trainer [28] on Pytorch [29] backbone using Paperspace service. This study applies different preliminary trials with different hyper-parameters to find the best initial learning rate value, batch size value, and the number of training epochs. The mode was fine-tuned on mMARCO dataset [30] and optimized using AdamW optimizer model.

4.2 Datasets

Dataset Preparation for Question Generation Task. The mMARCO [30] dataset is a machine-translated multilingual version of the MS MARCO passage ranking dataset covering 13 typologically diverse languages. We imported and filtered the dataset into Arabic language. The Arabic dataset has 24+ M query records, including question queries. The records that contained interrogative questions ('ما - What', 'كم - How much/many', 'أين - Where', 'ماذا - What', 'هل – Does /Do /Is /Are', 'كيف - How', 'لماذا - Why', 'من – Who /Whom', 'متى - When', 'إلى - To') were filtered. The column negative contains

different meanings or wrong sentences removed from the dataset. All the filtered texts are cleaned and saved as a new dataset, and we came up with 242,060 recodes. Following Zhao et al. [17], we split the dataset into (80%) for training, (10%) for evaluation, and (10%) for testing, and the total numbers for training, evaluation, and testing are 193648, 24206, and 24206, respectively.

Dataset Preparation for Sentence Extraction. The TyDi Qa is a question-answering dataset [31] with 204K question-answer pairs in 11 languages. It has 23K paragraphs with answers in Arabic language. The data was collected without translation in each language. We used the Arabic dataset, and it cleaned by removing duplicate contents and removing unwanted columns. We come up with 17804 articles.

4.3 Evaluation Metrics

Arabic Automatic Question Generation. We conduct experiments for Arabic automatic question generation task using mMARCO [30] dataset to generate a question according to the given input text without answer-aware. Predicting a question is formulated as a sequence-to-sequence (seq2seq) problem, while the first segment is the input text, and the second is the generated question. We also fine-tuned the model on the training dataset for 4 epochs with batch size value 8 and learning rate value 1e-5 using AdamW optimizer model. All the weights were initialized from AraBERT checkpoint, except the encoder-decoder attention variable was initialized randomly. We evaluated the AAQG base model using the stander BLEU [32], ROUGE [33] and METERO [34] to report the model results. They are the most common evaluation metrics for QG tasks.

Table 1. The proposed model evaluation results.

Model	Dataset	BLEU-4	METEOR	ROUGE-L	Language
Wang et al. [18]	SQuAD	14.78	18.61	41.87	English
UNILM$_{LARGE}$ Dong et al.[35]	SQuAD	22.12	25.06	51.17	English
ERNIE-GEN$_{BASE}$ Xiao et al. [15]	SQuAD	22.28	25.13	50.58	English
The Proposed Model-*base*	mMARCO	19.12	23.00	51.99	Arabic

We compare the performance of our model against several best QG models, including state-of-the-are in English language, because there are no Arabic baselines results to compare with. The results are presented in Table 1. Although our model generates questions without an answer, the other comparing models conduct experiments to generate questions with input text and a given answer for the QG task. Wang et al. [18] is based on a sequence-to-sequence model with an additional hidden pivot predictor to get the candidate's answer. UNILM*LARGE* [35] is based on a sequence-to-sequence model with unidirectional and bidirectional language models. ERNIE-GEN*BASE* [15] is based on a single multi-flow sequence to sequence with Noise-Aware generation. AAQG outperforms Wang et al. [18] model in all the evaluation metrics and exceeds all the other models in ROUGE-L metric.

Text Extraction. We evaluate the proposed method using the standard ROUGE metric [33], the most common automated measure in text summarization. The evaluation calculated the precision value, recall value, and F1-score for ROUGE-1, ROUGE-2, and ROUGE-L. Many researchers select random documents from the dataset to evaluate the method, and we evaluated the method using the first approach that Abu Nada et al. [36] used to evaluate their model. First, all the documents in the dataset were evaluated individually. Second, the documents were split into paragraphs, and all the split paragraphs were evaluated separately. The average results for the documents and paragraphs scores were calculated as the mothed accuracy and presented in Table 2. The results scores number rounded to two decimal places.

Table 2. The proposed method evaluation results using ROUGE-1, ROUGE-2, and ROUGE-L.

	Counts	ROUGE-1			ROUGE-2			ROUGE-L		
		P	R	F1	P	R	F1	P	R	F1
Documents	17804	0.63	0.57	0.91	0.55	0.50	0.80	0.63	0.57	0.90
Paragraphs	23911	0.61	0.57	0.82	0.48	0.45	0.66	0.61	0.57	0.82

The summary results in Table 2 show the average accuracy results of the documents compared to the paragraphs. They are close in the precision score and a difference in the F1-score of ROUGE-1 and ROUGE-L due to increased paragraphs count. On the other hand, the recall score of ROUGE-1 and ROUGE-L is the same nevertheless of the count of paragraphs, with 0.57 and 0.57, respectively. The proposed approach is to extract the key sentence from the paragraphs for the question generation task and to compare the accuracy of the paragraphs based on documents accuracy.

5 Conclusion and Future Work

We introduce an Arabic automatic question generation model that can generate N interrogative questions from a single unlimited-length document. It generates the question without answer-aware by splitting the document into paragraphs and extracting the key sentences. In addition, we also adopted the BERTShared encoder-decoder model for the Arabic language and fine-tuned it for the question generation task. In the future, we will investigate adapting a large network size to study how the size impacts the process of generating questions in Arabic. We also tend to generate various types of questions in Arabic language.

6 Acknowledgement

We would like to express our gratitude to Universiti Teknologi Malaysia (UTM) - research grant UTMER 19J10 for the direct and indirect support towards this research.

References

1. Patil, M.N., et al., A Survey on Automatic Multiple Choice Questions Generation from Text. 2021.
2. Chen, X. and J. Xu. An Answer Driven Model For Paragraph-level Question Generation. IEEE.
3. Zhang, R., et al., A Review on Question Generation from Natural Language Text. ACM Transactions on Information Systems, 2022. 40(1): p. 1-43.
4. Kurdi, G., et al., A Systematic Review of Automatic Question Generation for Educational Purposes. International Journal of Artificial Intelligence in Education, 2019. 30(1): p. 121-204.
5. Alazani, S.A. and C.N. Mahender. Rule Based Question Generation for Arabic Text. in Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence. 2021. ACM.
6. Bousmaha, K.Z., et al., AQG: Arabic Question Generator. Revue d'Intelligence Artificielle, 2020. 34(6): p. 721-729.
7. Elbasyouni, M., E. Abdelrazek, and A. Saad, BUILDING A SYSTEM BASED ON NATURAL QUESTION GENERATIO MohamedElbasyouni. 2014.
8. Vaswani, A., et al. Attention is all you need. in Advances in neural information processing systems. 2017.
9. Devlin, J., et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv pre-print server, 2019.
10. Etaiwi, W. and A. Awajan, Graph-based Arabic NLP techniques: a survey. Procedia computer science, 2018. 142: p. 328-333.
11. Alec Radford, J.W., Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, Language models are unsupervised multitask learners. Open AI Blog, 2019.
12. Raffel, C., et al., Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv pre-print server, 2020.
13. Varanasi, S., S. Amin, and G. Neumann. CopyBERT: A unified approach to question generation with self-attention. in Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI. 2020.
14. Liu, M. and J. Zhang, Chinese Neural Question Generation: Augmenting Knowledge into Multiple Neural Encoders. Applied Sciences, 2022. 12(3): p. 1032.
15. Xiao, D., et al., ERNIE-GEN: An Enhanced Multi-Flow Pre-training and Fine-tuning Framework for Natural Language Generation. arXiv pre-print server, 2020.
16. Fatih, et al., Automated question generation and question answering from Turkish texts using text-to-text transformers. arXiv pre-print server, 2021.
17. Zhao, Z., et al., Educational Question Generation of Children Storybooks via Question Type Distribution Learning and Event-Centric Summarization. arXiv preprint arXiv:2203.14187, 2022.
18. Wang, B., et al. Neural question generation with answer pivot. in Proceedings of the AAAI Conference on Artificial Intelligence. 2020.
19. Lopez, L.E., et al. Simplifying paragraph-level question generation via transformer language models. in Pacific Rim International Conference on Artificial Intelligence. 2021. Springer.
20. Wang, S., et al., A Multi-Agent Communication Framework for Question-Worthy Phrase Extraction and Question Generation. Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 33: p. 7168-7175.
21. Ghanem, B., et al., Question Generation for Reading Comprehension Assessment by Modeling How and What to Ask. arXiv preprint arXiv:2204.02908, 2022.

22. Gollapalli, S.D., et al., PaintTeR: Automatic Extraction of Text Spans for Generating Art-Centered Questions. 2022.
23. Wijanarko, B.D., et al., Question generation model based on key-phrase, context-free grammar, and Bloom's taxonomy. Education and Information Technologies, 2021. 26(2): p. 2207-2223.
24. Rothe, S., S. Narayan, and A. Severyn, Leveraging pre-trained checkpoints for sequence generation tasks. Transactions of the Association for Computational Linguistics, 2020. 8: p. 264-280.
25. Antoun, W., F. Baly, and H. Hajj, AraBERT: Transformer-based Model for Arabic Language Understanding. arXiv pre-print server, 2021.
26. Mihalcea, R. and P. Tarau. Textrank: Bringing order into text. in Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.
27. Elbarougy, R., G. Behery, and A. El Khatib, Extractive Arabic text summarization using modified PageRank algorithm. Egyptian informatics journal, 2020. 21(2): p. 73-81.
28. Wolf, T., et al. Transformers: State-of-the-art natural language processing. in Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. 2020.
29. Paszke, A., et al., Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 2019. 32.
30. Bonifacio, L.H., et al., mmarco: A multilingual version of the ms marco passage ranking dataset. arXiv preprint arXiv:2108.13897, 2021.
31. Clark, J.H., et al., TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. Transactions of the Association for Computational Linguistics, 2020. 8: p. 454-470.
32. Papineni, K., et al. Bleu: a method for automatic evaluation of machine translation. in Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.
33. Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. in Text summarization branches out. 2004.
34. Banerjee, S. and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. in Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. 2005.
35. Dong, L., et al., Unified Language Model Pre-training for Natural Language Understanding and Generation. arXiv pre-print server, 2019.
36. Abu Nada, A.M., et al., Arabic text summarization using arabert model using extractive text summarization approach. 2020.