End-to-End Quiz-Style Question Generation for Educational Purposes

2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS) | 979-8-3503-2210-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICICIS58388.2023.10391138

Mohammed Ashraf Mohammed
Computer Science Department Faculty of Computer and Information Science, Ain Shams University
Cairo, Egypt
mohammed20191700509@cis.asu.edu.eg

Mostafa Ashraf Borhamy
Computer Science Department Faculty of Computer and Information Science, Ain Shams University
Cairo, Egypt
mostafa20191700639@cis.asu.edu.eg

Mostafa Ayman Awad
Computer Science Department Faculty of Computer and Information Science Ain Shams University
Cairo, Egypt
mohammed20191700509@cis.asu.edu.eg

Mostafa Labib Mohamed
Computer Science Department Faculty of Computer and Information Science Ain Shams University
Cairo, Egypt
mostafa20191700646@cis.asu.edu.eg

Mostafa Hesham Abd-Elhassib
Computer Science Department Faculty of Computer and Information Science Ain Shams University
Cairo, Egypt
mostafa20191700656@cis.asu.edu.eg

Aisha Soliman Fath-Allah
Computer Science Department Faculty of Computer and Information Science Ain Shams University
Cairo, Egypt
aisha20191700330@cis.asu.edu.eg

Asmaa Bahai
Computer Science Department Faculty of Computer and Information Science Ain Shams University
Cairo, Egypt
Asmaa.bahai@cis.asu.edu.eg

Sally Saad
Computer Science Department Faculty of Computer and Information Science Ain Shams University
Cairo, Egypt
sallysaad@cis.asu.edu.eg

***Abstract*—Question Generation (QG) is a branch of Natural Language Processing (NLP). It aims to generate natural language questions based on given contents. By combining NLU and NLG, this paper aims to develop a fully functional question generation system which can be used in real-world applications for educational purposes. Crafting exam-style questions is an essential component of education, it serves diverse objectives where both the students and educators can use it to facilitate the education process. A pipelined system is introduced with two modules, an Answer Extraction module, and a Question Generation module. The Answer Extraction module is an encoder-decoder model that extracts question-worthy key-phrases from contexts. The Question Generation module consists of a set of specialized models finetuned on different types of questions that are generated in an end-to-end manner. The types of questions that can be generated by the proposed system are: WH, MCQ, complete, and true/false. The proposed comprehensive evaluation demonstrates the effectiveness of the system in generating educationally valuable question-answer pairs using only context paragraphs as input. This highlights the practical applicability of question generation techniques.**

Keywords— Natural language processing, Question generation, Question answering, Deep learning, Artificial intelligence

I. Introduction

Recent advances in Natural Language Processing (NLP) can be greatly attributed to the invention of the Transformer architecture [1]. Advancements in deep learning paved the way for the Transformer architecture. The Transformer architecture allowed development of models that have achieved state-of-the-art results on all NLP tasks, including question generation (QG). The original Transformer architecture consists of an encoder and a decoder connected to it. The encoder maps an input sequence of symbol representations $(x1, \ldots, xn)$ to a sequence of continuous representations $z = (z1, \ldots, zn)$. Given z, the decoder then generates an output sequence $(y1, \ldots, ym)$ of symbols one element at a time. At each step the model is auto regressive, consuming the previously generated symbols as additional input when generating the next. A significant advancement in the Transformer architecture is self-attention, which allows the model to simultaneously analyze the entire input sequence while processing each word, leading to a more refined encoding. All the models in the proposed system adopt the Transformer architecture for the task of end-to-end question generation. The proposed system is a pipelined system composed of two modules, an Answer Extraction module, and a QG module (Fig. 1). It is to be claimed that the proposed system can be used for educational purposes to create exercises. In turn it can be used to generate exercises on teaching material as they may not always be available, which puts a lot of pressure on both educators and students. For educators, this pressure comes from manually constructing questions which is a complex task that demands expertise, practice, and adequate resources. For students, this pressure comes from summarizing studying material and finding questions on it. These tasks are time-consuming therefore automating them would save time and money.

The goal of the Answer Extraction module is to train a neural model that when given a paragraph as input, outputs phrases in that paragraph that can be answers to questions. The goal of the model is to extract question-worthy key phrases that when given to a QG model will result in generating a question that is not only correct syntactically but also

979-8-3503-2210-1/23/$31.00@2023 IEEE

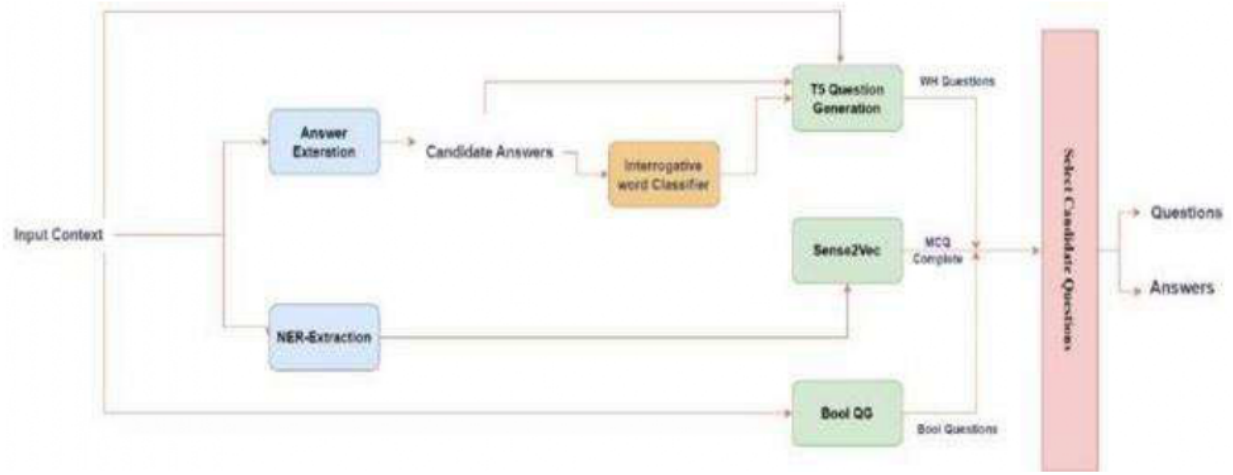

Fig. 1 Overview of the Question Generation pipeline.

semantically. The Answer Extraction model utilizes the powerful T5 [2] model to extract key phrases. The T5 model is based on the Transformer architecture, which is a neural network architecture that is well-suited for NLP tasks. The T5 model utilizes a text-to-text format, meaning that both the input and output are text. The goal of the QG module is to finetune a set of QG models for generating different types of questions. These types are as follows: WH, MCQ, complete, and true/false. Each model takes a passage, and an answer as input and generates high-quality, contextually relevant questions. Additionally, The WH-QG model is enhanced by utilizing a BERT-based classifier to predict the interrogative word [3].

Given a passage–answer pair the classifier predicts the interrogative word among seven classes. By incorporating the T5 model in the proposed pipeline, which has shown remarkable performance in NLP tasks, The aim is to enhance the QG process and provide accurate, informative, and diverse questions.

II. RELATED WORKS

Historically, rule-based systems have been used for QG. They are a type of Artificial Intelligence (AI) that uses templates and heuristics to generate questions. The rule-based approaches typically include the following steps: 1) Preprocessing of the input text using NLP techniques. 2) Identifying the words or phrases that will be considered an answer by using semantic roles and rules. 3) Using a set of predefined rules and templates, questions are generated. 4) Lastly, the generated questions are ranked by using a set of features. There are multiple disadvantages to using rule- based systems which are as follows: 1) The cost of creating the templates and rules manually is high. 2) Both the answers, and templates are not diverse as they depend on a set of rules. The question generation models mostly use neural-based networks that utilize a sequence-to-sequence architecture to craft questions about text in a natural language manner. In the Interrogative-word- Aware Question Generation (IWAQG) [3] work, a sequential system consists of two components: the first model predicts the appropriate interrogative word, which is passed to the second model to generate the corresponding questions. This allows the QG model to be able to focus more on the rest of the question rather than the question word.

Building upon previous work, the authors in [4] introduce a novel neural question generation (QG) model that emphasizes both answer relevance and positional awareness. To ensure that the generated question word aligns with the answer type, the model incorporates answer embedding, thereby explicitly considering the answer type during question word generation. Additionally, the model employs positional awareness, allowing it to consider the relative distance between the answer and the passage when selecting words to the question. In this study [5] explore the interconnectedness of question generation (QG) and question answering (QA), proposing that their inherent relationship can be exploited to enhance performance and reduce the reliance on labeled data, particularly in training a QA system.

Their contributions can be summarized as follows: 1) Utilizing question generation by integrating GPT-2 and BERT in an end-to-end trainable framework, enabling semi-supervised learning. 2) Employing QA as a proxy measure for evaluating question generation quality. In [6], the QG task is examined from the perspective of Knowledge Graph (KG) queries. Essentially, the QG task represents the capacity of intelligent systems to transform schema specific and abstract knowledge graph representations into natural language questions. The paper delves into complex question generation from a knowledge graph, by leveraging existing simple questions.

In this study [7], a clue word predictor is introduced, which predicts potential clue words for target questions based on the input passage context. To understand the relation between the answer and other tokens in the context, the predictor employs a syntactic dependency tree. For clue word sampling, it employs a Straight-Through Gumbel-SoftMax estimator and a GCN-based encoder. The predicted clue word distribution is then fed into the passage encoder, which includes a low-frequency masking technique to improve the performance in tuning input word embeddings. Using a multitask learning strategy, the decoder learns word generation probabilities from the vocabulary as well as word copying from the input passage. The copy gate in the model employs explicit binary labeling where the target vocabulary is further shortened based on the frequency distribution of non-overlapping words.

III. THE PROPOSED MODEL

A. Problem Statement

The problem can be divided into two subproblems: First, given passage P, a set of questions are required to be extracted-worthy key-phrases K. More formally:

$$\bar{K} = \text{argmax}_k \text{Prob}(K|P) \quad (1)$$

Second, having input passage $\boldsymbol{P}$, and an answer $\boldsymbol{A} \in \bar{\boldsymbol{K}}$ The aim is to find a question $\mathbf{Q}$ whose answer is A. Additionally, $\boldsymbol{Q} \in \boldsymbol{G}$ where $\mathbf{G}$ is the set of types of questions the proposed system can generate. More formally:

$$\bar{Q} = \text{argmax Prob}\left(Q_q|P, A\right) \quad (2)$$

Where $\boldsymbol{P}$ is a paragraph consists of a collection of words: $P = \{x_t\}^M$, and the answer is a sub span of $\boldsymbol{P}$. The problem is modeled in a pipelined manner consisting of two modules.

The first module is the Answer Extraction module which aims to solve the first subproblem. The second module is the QG module which aims to solve the second subproblem.

B. Answer Extraction

Most literature in the field of QG does not tackle the task of extracting key-phrases although it enables the QG models to generate meaningful questions. Often third-party libraries and automatic key-phrase extraction methods will be employed. Automatic key-phrase extraction methods are divided mainly into two categories: 1) statistical-based methods such as TF-IDF, and YAKE. 2) graph-based methods such as Text Rank, and RAKE. Although these methods are good at extracting key-phrases that summarize text, they do not always satisfy the requirement of extracting key-phrases which are question-worthy. In some literature in QG the answer will be assumed to be a given input to the QG model. This has the downside of not being applicable to real- life scenarios. It is believed thattraining a dedicated model for this task can yield better results for the quality of the generated questions. A dedicated model is argued tolearn the semantics of the passage and extract human-like answers from the text. Additionally, since the proposed model generates a set of answers from a context, it eliminates the need for a heuristic for the number of key-phrases to extract, thus reducing the number of illogical questions.

The answer extraction model utilizes the powerful T5 model to extract key phrases. The T5 model has several unique features that made it a fit for the proposed task. First, the T5 model is pretrained on a massive dataset of text (C4 dataset) [2], which allows it to learn a wider variety of relationships between words. Second, the T5 model can learn the context of a phrase, which helps it to identify phrases that are important even if they do not appear frequently. These features enable it to achieve good performance when it comes to the task of extracting question-worthy key phrases.

C. WH-Question Generation

1) Interrogative word Classifier

In the proposed WH-question generation approach, an interrogative word classifier is utilized based on BERT. The passage and the extracted answer from the Answer Extraction module are provided as input. To distinguish the answer span and highlight its importance, a special token is incorporated, [ANS], which informs BERT about its distinct significance compared to the remainder of the passage. BERT outputs a special token, [CLS], originally designed for classification tasks. A feed-forward network is constructed on the top of BERT that takes the concatenated embeddings of the [CLS] token, and a trainable embedding representing the type of the entity of the answer as input. The interrogative-word classifier is connected to the second module by using the generated interrogative word as the first input to the WH-QG model.

2) Question Generation

A robust T5-based model is utilized to generate questions. This model receives as input a passage, an answer, and an interrogative word, and it produces high-quality questions that are relevant to the given context. By incorporating the T5 architecture, which has demonstrated exceptional performance in various NLP tasks, the aim is to improve the QG process and deliver precise, informative, and diverse questions. Since the QG module employs a T5. model, both the input and the output are in text format. The following is an example input provided to the model:

<interrogative word> Answer: <answer> Context: <context>

D. Boolean Question Generation

Boolean question generation is the task of generating questions for which the answers are either true or false. This task differs from the task of generating WH-style questions. The argument is that the structure of the question significantly differs from the typical WH-style question. In WH-style question there exists an answer, based on which, the question is generated. On the other hand, for Boolean QG the answer is either true or false. This makes choosing a suitable sentence for generating a question a complex task. For this reason, a separate model is finetuned for this task. T5 is chosen for this task.

E. MCQ & Complete

MCQ and complete are related tasks, in complete the task is to extract a key phrase and its sentence omitting the key phrase (answer), for MCQ the task of generating distractors related to the answer is added. MCQ and complete modules share the same steps except that MCQ gives you multiple possible answers in the form of distractors. The distractors in MCQ are generated by the Sense2Vec [8] model. Sense2Vec is an extension of Word2Vec. Unlike Word2Vec, which generates embeddings for individual word tokens, Sense2Vec creates embeddings for "senses" of words. A sense is a combination of a word and a label that represents the specific context in which the word is used. This label can indicate various aspects such as part-of-speech (POS) tags, polarity, entity names, or dependency tags. For example, in Word2Vec you can have the following:

$$v[king] - v[man] + v[woman] \approx v[queen] \quad (3)$$

where $\boldsymbol{v}$ denotes vector embedding of word. In Sense2Vec POS tags as well as entity names are used to generate word embeddings. We take advantage of spaCy [9] being able to support adding Sense2Vec as part of a pipeline and use this pipeline to generate MCQ and complete questions.

IV. EXPERIMENTS

A. Datasets

Four reading comprehension datasets were used in the training the proposed models as follows:

- Stanford Question Answering Dataset (SQuAD) [10] is a collection of questions created by crowd workers based on various Wikipedia articles. In this reading comprehension dataset, each question has a specific answer, which is a segment of text or span taken from the related passage in the article.
- NewsQA [11] is a demanding machine comprehension dataset comprising more than 100,000 question-answer pairs generated by humans. Questions and answers are provided by crowd workers who utilize a pool of over 10,000 news articles from CNN. The answers are segments of text extracted from the relevant articles.
- AdversarialQA [12] consists of three recently developed reading comprehension datasets, created through the incorporation of an adversarial model in the construction process. The authors employed three distinct models in the annotation loop to generate three separate datasets. Each dataset comprises 10,000 training examples, along with 1,000 examples for both validation and testing.
- BoolQ [13] is a dataset designed for answering yes/no questions and comprises 15,942 examples. The questions in this dataset are naturally occurring, meaning they are generated in spontaneous and unrestricted settings.

As can be seen the datasets are from different sources which introduces variety in the data that should help the models learn and generalize on unseen data examples.

For Answer Extraction SQuAD and NewsQA were used. Additionally, preprocessing was applied to group all answers of a unique context into a single record. After appending all the answers of a context with a special separator token, the text is passed as the output in the training of the proposed Answer Extraction model. For WH question generation all datasets were used except for BoolQ. The distribution of interrogative words within the datasets is not balanced. the dataset was down sampled to create a balanced dataset that used for the training of Interrogative-Word Classifier. This down-sampling process ensures an equal representation of different interrogative words in the training data for effective training of the classifier. 10000 samples were sampled for each class except for "why" type which was around 2700 samples. For Boolean QG BoolQ was used.

B. Implementation

1) Answer Extraction

The proposed Answer Extraction model is a T5 model, specifically a t5-small model. The t5-small model is provided by HuggingFace [14]. The model was trained for 8 epochs each is with batch size four. The first epoch being a warmup epoch where the learning rate starts increasing gradually until reaching 3e-4. For the remaining 7 epochs a polynomial decay function of power 1 is applied to decrease the learning rate gradually until it reaches 3e-5 at the end of the last epoch. The optimizer used for training is the Adam optimizer and the loss function is the cross-entropy loss function. The size of the encoder was set to 250, and the size of the decoder was set to 70. The task prefix used was "extract answers".

2) WH-question Generation

An interrogative word classifier was developed for the proposed system using BERT-base-uncased, a pre-trained model provided by HuggingFace. The objective function used by the classifier was cross-entropy. The classifier trained for 2 epochs. To obtain entity types embedding, spaCy [9] is utlized, which provides six dimensions: PERSON, CARDINAL, DATE, ORG, GPE, and NONE (when no entity type is determined). The input dimension of the Feed-Forward Network (FFN) is 773, consisting of 768 dimensions from BERT and 5 dimensions from the entity type embedding. The FFN predicts the interrogative words: what, who, which, where, how, when, why, and others, and its output dimension is 8. The FFN architecture consists of a SoftMax layer stacked over a single linear layer. For optimization, the Adam optimizer is used with weight decay and set the learning rate to 5e-5.

The t5-base model is utilized for the proposed QG model. The task prefix used was "generate question using question word". During training, the learning rate is set to 5e-5 and the encoder size to 250 tokens. For decoding, a size of 70 tokens is used. The QG model was trained for three epochs, allowing it to learn and improve over multiple iterations.

3) Boolean Question Genration

The Boolean question generation model uses the HuggingFace TensorFlow implementation of the t5-base model. The model was trained with Adam optimizer with weight decay using the cross-entropy loss function. The model was trained for 15 epochs with a constant learning rate of 3e-4 and a batch size of 8. The size of the encoder was set to 500, while for the decoder the size was set to 70. The task prefix used was "generate Boolean question".

4) MCQ & Complete

Sense2Vec is used as part of a spaCy pipeline to generate questions. Sense2Vec can extract the sense of the word on its own, but its accuracy was not good enough for us. The "en_core_web_lg" spaCy model is exploited for the proposed pipeline. Sense2Vec is pretrained on two datasets of Reddit comments, the first being in 2015, and the second being in 2019. The 2019 dataset is used as it can capture the most recent meanings of words for example the word ghost can mean a supernatural spirit (2015 dataset) or the action ignoring your spouse (2019 dataset). SpaCy is used to extract all named entities in the input text. Several filtering techniques are applied to enhance the quality of the distractors. The first technique is to make sure that the NER tag of the distractor matches that of the correct answer. The second technique is making sure there are no common words between the distractor and the correct answer. This is done as the dataset is collected from comments on the Reddit website which are prone to spelling mistakes. As such the same word with a different spelling can have two different embeddings where similarity between them is high leading to multiple correct answers or duplicate distractors. The third technique is making sure that cardinals in different formats are not chosen as distractors e.g., if "1" is the answer then "one" is filtered from being chosen as a distractor. The complete questions are the same as MCQ questions but without any distractors and the correct answer is omitted from the text.

V. Evaluation & Results

A. Answer Extraction

The evaluation methodology proposed in [15] was useddue to its close resemblance to the required task. The authors extended the SQuAD F1 metric, originally devised for a single answer span, to encompass multiple spans within a document, introducing the multi-span F1 score. This metric is calculated by constructing a pairwise, token-level F1 score matrix of elements $f(i,j)$ between a predicted phrase $\hat{e}i$ and a gold phrase $\boldsymbol{e_j}$. Max-pooling along the gold-label axis effectively assesses the precision of each prediction, partial matches accounted for by the pairwise F1 (identical to evaluation of a single answer in SQuAD) in the cells:

$$p_i = \max(f_{i,j}) \tag{4}$$

Analogously, the recall for label $\boldsymbol{e_j}$ can be computed by max-pooling along the prediction axis:

$$r_j = \max(f_{i,j}) \tag{5}$$

The multi-span F1 score using the mean of Precision is defined as:

$$\bar{p} = avg(p_i) \tag{6}$$

and recall:

$$\bar{r} = avg(r_i) \tag{7}$$

So F1 is:

$$F1_{MS} = \frac{2\bar{p} \cdot r}{\bar{p} + \bar{r}} \tag{8}$$

The proposed baseline model is PtrNet proposed in [15]. PtrNet is a pointer network which uses RNN encoder and decoder that points to key-phrase start and end boundaries. Another baseline, YAKE is used, which is automatic key-phrase extraction method based on statistical methods. All the key phrases are marked as answers. Additionally, the proposed model is compared with a simple method which marks all named entities as answers using spaCy library. The results can be seen in Table I. The proposed model outperforms all the baseline models in the F1 score when it comes to the SQuAD dataset. NewsQA dataset in the proposed model is observed to outperform both the spaCy and YAKE but is exceeded in evaluation by PtrNet. This is believed to be due to NewsQA having more answers per unique context than SQuAD. Additionally, the lengths of contexts are significantly bigger in NewsQA which makes only part of the context visible to the encoder of the proposed model. Since SQuAD is collected from Wikipedia articles and NewsQA is collected from CNN articles, SQuAD performance is believed to be more relevant when it comes to educational settings. This is because text in educational settings is more like that of Wikipedia articles than CNN articles. Therefore, the lack of performance when it comes to NewsQA is believed not to be representative of the model's ability to extract question- worthy answers.

TABLE I. Comparison between benchmark and the proposed Answer Extraction model denoted by "*"

Model	F1	Precision	Recall
SQuAD			
spaCy NER	26.4%	30.7%	23.2%
YAKE	23.8%	28.6%	20.3%
PtrNet	40.4%	**44.8 %**	38.7%
T5-small*	**45.4%**	44.3%	**46.6%**
NewsQA			
spaCy NER	10.1%	12.5%	8.5%
YAKE	24.0%	28.7%	20.7%
PtrNet	**43.5%**	**46.7%**	**42.7%**
T5-small*	39.1%	36.9%	41.7%

B. WH-question Genration

1) Interrogative Word Classifier

Classification accuracy is used to evaluate the proposed Interrogative- Word Classifier. The proposed model is compared with the Interrogative-Word Classifier proposed in [3]. The Interrogative-Word Classifier is a BERT based classifier that uses context and answer NER tag embedding to make a classification decision. As can be seen in Table II the proposed model achieves the same results as the baseline model.

2) Question genration

The following metrics are used for evaluating the proposed WH QG model: BLEU1:4, ROGUE-L, and METEOR. The Interrogative-Word-Aware Question Generation (IWAQG) model [3] is used as a baseline. IWAQG is based on a sequence-to-sequence architecture neural based network that utilizes a gated self-attention in the encoder and an attention mechanism with maxout pointer in the decoder. Additionally, the model utilizes the interrogative word classifier described in Section III to choose the interrogative which is prepended to the input text. The Clue Guided Copy Network for Question Generation (CGC-QG) model [7] is also used as a baseline. The clue word predictor of CGC-QG uses a syntactic dependency tree to predict potential clue words for questions based on passage context. It employs a GCN-based encoder and a Gumbel-SoftMax estimator for clue word sampling. The predicted clue word distribution is used in the passage encoder with a low- frequency masking strategy. The decoder uses multitask learning to learn word generation probabilities and word copying from the passage, employing a copy gate, and shortening the target vocabulary based on non-overlapping word frequencies. As can be seen in Table III, the proposed model outperforms the baselines in all metrics.

C. Boolean Question Generation

For evaluating the proposed model, the following metrics BLEU1:4, ROUGE-L, and METEOR are used. The proposed model is compared with the Finetuned-QG model [16]. Finetuned-QG is a T5 QG model that has been trained on multiple QG datasets with different formats in a transfer learning manner. Finetuned-QG is the authors' best performing model in Boolean QG hence being chosen as a baseline model. As can be seen in Table IV, the proposed model outperforms Finetuned-QG on all metrics except for a 0.4 difference in ROUGE-L metric in favor of Finetuned-QG.

TABLE II. COMPARISON BETWEEN BENCHMARK AND THE PROPOSED INTERROGATIVE WORD CLASSIFIER MODEL DENOTED BY "*".

Model	Accuracy
Interrogative aware classifier	**73.8%**
Bert-base*	72.7%

TABLE III. COMPARISON BETWEEN BENCHMARKS AND CHOSEN WH QG MODEL DENOTED BY "*"

Model	BLEU- 1	BLEU- 2	BLEU- 3	BLEU- 4	METEOR	ROGUE- L
Finetuned - QG	49.49	33.70	24.71	18.51	23.68	**47.04**
T5-base*	**50.05**	**37.86**	**29.14**	**22.60**	**24.80**	46.62

TABLE VI COMPARISON BETWEEN BENCHMARK AND THE PROPOSED BOOLEAN QG MODEL DENOTED BY "*".

Model	BLEU- 1	BLEU- 2	BLEU- 3	BLEU- 4	METEOR	ROGUE- L
SQuAD						
IWAQG	47.69	32.24	24.01	18.53	22.23	46.94
CGC-QG	40.45	23.52	15.68	11.06	17.11	43.16
T5-base*	**49.59**	**33.79**	**25.27**	**19.59**	**24.51**	**47.36**

VI. CONCLUSION & FUTURE WORK

In this work, a pipelined system for end-to-end question generation was proposed. This system consists of two modules, an Answer Extraction module, and a question generation module. The first module is used to extract question-worthy key phrases from text. The second module is comprised of a set of finetuned models that generate the following types of questions: WH, true/false, MCQ, and complete. The proposed proposed system generates high quality, contextually relevant questions that can be used for educational purposes. The proposed hypothesis was proven by conducting quantitative analysis on all the proposed models which show that the models of the proposed system outperform powerful baseline models and are viable for educational settings.

In the future, we would like to improve the proposed Boolean QG model by creating a new dataset. We believe this can be achieved by applying the methods used in the MCQ and Complete module on the contexts of a reading comprehension dataset and training the model on the new dataset. Furthermore, making the proposed system not only limited to the English language by using multilingual models and datasets should prove beneficial for the domain of education.

REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez,

[2] A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[3] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M.,

[4] ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), 5485-5551.

[5] Kang, J., Roman, H. P. S., & Myaeng, S. H. (2019). Let me know what to ask: Interrogative-word-aware question generation. arXiv preprint arXiv:1910.13794.

[6] Sun, X., Liu, J., Lyu, Y., He, W., Ma, Y., & Wang, S. (2018). Answer-focused and position-aware neural question generation. In Proceedings of the 2018 conference on empirical methods in natural language processing (pp. 3930-3939).

[7] Klein, T., & Nabi, M. (2019). Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. arXiv preprint arXiv:1911.02365.

[8] Zhao, J., Deng, X., & Sun, H. (2019). Easy-to-hard: Leveraging simple questions for complex question generation. arXiv preprint arXiv:1912.02367.

[9] Liu, B., Zhao, M., Niu, D., Lai, K., He, Y., Wei, H., & Xu, Y. (2019, May). Learning to generate questions by learningwhat not to generate. In The world wide web conference (pp. 1106-1118).

[10] Trask, A., Michalak, P., & Liu, J. (2015). sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. arXiv preprint arXiv:1511.06388.

[11] Honnibal, M. (2015). Spacy · industrial-strength natural language processing in python. · Industrial-strength Natural Language Processing in Python. https://spacy.io/.

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

[13] Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., & Suleman, K. (2016). Newsqa: A machine comprehension dataset. arXiv preprint arXiv:1611.09830.

[14] Bartolo, M., Roberts, A., Welbl, J., Riedel, S., & Stenetorp, P. (2020). Beat the AI: Investigating adversarial human annotation for reading comprehension. Transactions of the Association for Computational Linguistics, 8, 662-678.

[15] Clark, C., Lee, K., Chang, M. W., Kwiatkowski, T., Collins, M., & Toutanova, K. (2019). BoolQ: Exploring the surprising difficulty of natural yes/no questions. arXiv preprint arXiv:1905.10044.

[16] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (pp. 38-45).

[17] Subramanian, S., Wang, T., Yuan, X., Zhang, S., Bengio, Y., & Trischler, A. (2017). Neural models for key phrase detection and question generation. arXiv preprint arXiv:1706.04560.

[18] Yuan, W., Yin, H., He, T., Chen, T., Wang, Q., & Cui, L. (2022, April). Unified question generation with continual lifelong learning. In Proceedings of the ACM Web Conference 2022 (pp. 871-881).