

Development of system for generating questions, answers, distractors using transformers

Alibek Barlybayev^{1,2}, Bakhyt Matkarimov¹

¹Department of Artificial Intelligence Technologies, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

²Higher School of Information Technology and Engineering, Astana International University, Astana, Kazakhstan

Article Info

Article history:

Received Aug 2, 2023

Revised Oct 13, 2023

Accepted Dec 5, 2023

Keywords:

Automated test set generation

Multiple-choice question

Natural language processing

Question generation

Transformers

ABSTRACT

The goal of this article is to develop a multiple-choice questions generation system that has a number of advantages, including quick scoring, consistent grading, and a short exam period. To overcome this difficulty, we suggest treating the problem of question creation as a sequence-to-sequence learning problem, where a sentence from a text passage can directly mapped to a question. Our approach is data-driven, which eliminates the need for manual rule implementation. This strategy is more effective and gets rid of potential errors that could result from incorrect human input. Our work on question generation, particularly the usage of the transformer model, has been impacted by recent developments in a number of domains, including neural machine translation, generalization, and picture captioning.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Bakhyt Matkarimov

Department of Artificial Intelligence Technologies, L.N. Gumilyov Eurasian National University

11 Pushkin Street, Baykonur District, Astana, Kazakhstan

Email: bakhyt.matkarimov@gmail.com

1. INTRODUCTION

Question generation has become a popular trend in recent years and is being used for various applications, especially in education. Its main purpose is to generate natural questions from a given text, this can help students learn and understand reading materials better [1]. Test questions are an essential part of the learning process, helping to measure student understanding [2], [3]. Crafting and evaluating such questions can be a tedious and drawn out of activity, eating up a lot of time [4]. Consequently, researchers and tutors are extremely attracted to the idea of automatically generating questions and evaluating answers [5], [6]. Schools and universities usually conduct tests where students are required to pick the right answer from several options or fill missing words. To assess knowledge, multiple-choice questions (MCQ), true/false (T/F) and fill-in-the-blank (FiB) are widely used tools [7].

Question generation techniques mostly use of heuristics to convert descriptive text into corresponding question. Current rule-based methods divided into 3 broad categories: template-based [8] methods, syntax-based [9]–[11] approaches, and semantic-based [12]–[15] technologies. In essence, two primary steps required to successfully generate a response through AI-driven methods-context selection and question construction. These processes can be achieved by applying a semantic or syntactic parser to the text of an input context, enabling the algorithm to identify relevant topics that asked about. By taking into consideration the topic in the context, the intermediate representations converted to a natural language question. That is done either through a transformation-based approach or via templates. AI-driven processes are often dependent on manual feature engineering, a labor-intensive task that needs a lot of domain-specific knowledge and experience. These methods also comprise multiple components that lack scalability and reusability, making them less reliable.

There has been a sharp surge in deep neural models for the purpose of question generation. Such models are full data-driven and end-to-end trainable, affording the training of question construction and context selection to be undertaken simultaneously. Neural question generation models have proven to be more superior to rule-based methods. They produce better phrased and varied questions. Generating questions typical involves an approach called sequence-to-sequence (Seq2Seq). This method involves various encoders and decoders that help to produce higher quality questions. Without putting aside any potential approaches, this is the most common type of neural network used for question generation. In study [16], the first neural question generation model was introduced, which has shown to be much more effective than traditional rule-based methods as it uses recurrent neural network (RNN) based Seq2Seq model with attention [17]. Subsequent articles have tried to enhance the effectiveness of RNN-based Seq2Seq structures by using question types [18], [19], response position characteristics [20], [21], response splitting [22], [23] and implementing an internal attention mechanism [24], [25]. Question generation is gaining much attention, with popular frameworks like pre-trained framework [26], variational autoencoder [27], graph-based framework [28], and adversarial network [29] becoming increasingly popular. Maximum likelihood estimation is a widely used training strategy, but there are other options available. Multi-task learning [30], reinforcement learning [31], and transfer learning [32] have all proven to be effective in optimizing neural question generation models.

This article aims to create a MCQ generation system which offers several benefits, such as quick scoring, standardized grading and minimized examination duration. MCQ format has been proven advantageous in [33]. With numerous competitive exams available, MCQ have become the preferred assessment method to test a candidate's knowledge. Kazakhstan has implemented the unified national testing that based on these MCQs for university admissions. In addition, research confirms that MCQ is effective for use in higher education environments [34].

The design of MCQs [35] comprises three essential elements: the interrogative sentence, which sets the context or pose the question; the correct answer key, denoting the accurate response; and distractors, which are misleading options meant to challenge the test taker. In the realm of MCQs, the interrogative sentence serves as the foundation, often framing a problem or inquiry for the test taker. This question stem may feature a blank space or a direct question, prompting careful consideration of the available choices. The correct answer key in a multiple-choice question is pivotal, representing the option that aligns with the intended response to the question or scenario presented in the interrogative sentence. Distractors are a crucial aspect of MCQs, strategically crafted to resemble plausible answers. These incorrect choices aim to perplex the test taker, making it imperative to thoroughly evaluate each option in relation to the question stem. Effective MCQs employ a well-crafted interrogative sentence, ensuring that it engages the test taker and conveys the question clearly, even with a blank space for the answer. Additionally, a well-defined answer key and carefully constructed distractors are vital components in evaluating the test taker's comprehension and critical thinking abilities.

Constructing a well-structured MCQ necessitates a keen understanding of the types of sentences that lend themselves well to this assessment format [36]. An integral part of creating effective MCQs involves the careful selection of sentences from a given text, prioritizing those that convey the most crucial information. Various methodologies, outlined in academic literature, shed light on techniques to identify sentences best suited for MCQs, ranging from analyzing sentence length [37] to considering the presence of particular words [38] or parts-of-speech patterns [39]. Summarization techniques [40] and syntactic analysis [36] also offer valuable approaches to pinpointing sentences that are rich in informational content, ensuring MCQs are well-founded and meaningful. The informed choice of sentences for MCQs can greatly impact the effectiveness of the assessment, emphasizing the importance of considering diverse strategies, including sentence length, vocabulary, parts-of-speech, summarization, and syntax [36]–[40].

When constructing answer keys, it is vital to carefully consider which words will be replaced or removed from a sentence in order to create an interrogative phrase. This decision-making process requires skill and attention to detail [36]. Term frequency (TF) is a simple yet effective strategy of discovering the main subject in a sentence [41]. In certain circumstances, term frequency-inverse document frequency (TF-IDF) is utilized as an option to term frequency [42]. Various techniques have been proposed in the literature for choosing the correct answer to MCQs, such as part-of-speech matching [43], parse structure [44], pattern matching [44] and semantic information [45].

Once a keyword is chosen from an informative sentence, the next crucial step involves transforming it into a well-constructed interrogative sentence, forming the basis of an effective MCQ. The transformation of a selected keyword from an informative sentence into an interrogative sentence is a pivotal stage in crafting meaningful MCQs. Crafting a well-structured interrogative sentence based on a chosen keyword from an informative statement is an essential part of the process when generating stems for MCQs. Numerous methodologies, outlined in academic literature, offer valuable insights into creating effective

interrogative stems for MCQs, utilizing techniques such as dependency structure [45], wh-words [46], discourse connectives [47], and semantic information [48]. Exploring various approaches, such as employing wh-words [46] or considering dependency structures [45], plays a vital role in devising appropriate interrogative stems for MCQs, enhancing the overall quality of the assessment.

Poorly designed distractors in MCQs can negatively affect the quality of testing [49], as it becomes too easy to identify the correct answer. Hence, ensuring the distractors provided are of high quality is crucial for preserving the integrity of MCQs. If not, it could significantly reduce the effectiveness of testing. Various techniques such as parts-of-speech analysis [50], word frequency counting [41], WordNet [51], domain ontology [52], distributional hypothesis [45] and semantic analysis [53], [54] are being implemented in the current research to produce effective distractors for multiple choice questions (MCQs).

Crafting effective MCQs requires concise, simple sentences. To address this challenge, we propose the question generation problem should be treated as a sequence-to-sequence learning problem, meaning a sentence from a text passage can be mapped directly to a question. Our strategy is driven by data, eliminating the need for manual implementation of rules. This approach is more efficient and eliminates potential errors that may arise from inaccurate manual input. Recent progress in various areas, such as neural machine translation [17], [55], generalization [56], [57] and image captioning [58], has influenced our work on question generation—particularly through the use of the transformer model [59].

This article offers a comprehensive perspective to the existing literature, largely due to its essential features: i) We have designed a comprehensive system for the automated production of MCQs. That includes constructing a relevant question sentence, researching an answer key and formulating plausible distractors from text material for examination; ii) Thanks to its use of named entity recognition, this system is able to produce multiword distractors, making it very appealing; iii) Our question generation system showed the best automatic score among various question generation systems; iv) We compared the results of our model with the generative pre-trained transformer (GPT) technology. In terms of generating responses, GPT-model and our model give very similar results.

2. METHOD

2.1. Data set collection

To train a neural model we need to get question and answer inputs. There are a large number of publicly available question and answer datasets [60]. The AI2 reasoning challenge (ARC) dataset includes 7,787 multiple-choice science questions that created for grade-school level students [61]. It divided into two sets: challenge and easy. With this dataset, artificial intelligent (AI) reasoning can test and improved further. The challenge set is designed to include only the questions which both retrieval-based and word co-occurrence algorithms failed to answer correctly. Models' performance is evaluated by how accurate they are. Shaping answers with rules through conversation (ShARC) is a tricky question answering (QA) dataset that demands rational thinking, entailment/natural language interface (NLI) components and natural language generation [62]. Notably, the majority of machine reading research concentrates on question answering problems where the response can be found straight in the document to read. Yet, real-world question answering scenarios often involve reading a text not to explicitly identify the answer, but rather to understand how to use background knowledge to generate an answer. One example is the ShARC dataset contain more than 32,000 tasks. This dataset is quite demanding yet rewarding. The CliCR dataset composed almost 100,000 queries and corresponding documents which sourced from clinical case reports [63]. It tests the ability of readers to answer the query by providing a medical problem/test/treatment entity. Bridging inferences and tracking objects appear to be the essential abilities needed for effective answering. Such abilities frequently requested among those seeking successful results. The CNN/Daily Mail dataset is an ideal resource for those looking to develop skills in the area of Cloze-style reading comprehension [64]. The data was gathered from news articles on CNN and Daily Mail utilizing certain heuristic guidelines. Close-style questions involve using context clues to infer the answer. That involve creating the questions by replacing entities with an entity marker (@entityn) from bullet points summarizing aspects of the article. Coreferential entities, in particular, are replaced with a unique index (n).

We are testing the capacity of a model to detect missing information in bullet points based on the text from their respective articles. The results of the models evaluated through accuracy tests on test sets. CoQA is a massive dataset used for developing conversational question answering systems [65]. It has more than 127,000 questions and answers from 8,000+ conversations. The information was gathered by connecting two crowd workers who discussed a passage through questions and answers. HotpotQA is an impressive dataset with 113,000 Wikipedia-based question-answer pairs [66]. The questions posed by this dataset require finding and considering multiple related documents and are not limited to just one knowledge base. Additionally, sentence-level supporting facts for each question supplied as well. Microsoft AI and Research have created MS MARCO, a dataset that aimed at providing machine reading comprehension [67]. This

dataset consists of questions from actual user inquiries and answers which are generated by humans. Advanced search technology from Bing utilized in order to extract context passages from multiple, real documents. This data set contains an extensive amount of queries, 100,000, and a subset that feature multiple answers. MultiRC is a dataset consisting of short paragraphs and multi-sentence questions [68]. These questions can all be answered by referring to the given paragraph, making it ideal for testing natural language processing systems. The Natural Questions dataset holds questions taken from real-world users and put to the Google search engine [69]. For answer these, QA systems need to read and comprehend an entire Wikipedia article that could have, or have not the correct response. Whenever someone answers a question, a Wikipedia page accompanied by a long answer (normally a passage) and a short answer (one or more entities) will be shown. If there is no long/short answer present, it will be marked as "null". NewsQA is an extensive reading comprehension dataset derived from CNN's news articles [70]. It houses more than 100,000 human-generated question-answer pairs and spans of text across over 10,000 news stories. This dataset provides a powerful resource for building AI models to understand context. QAngaroo has two distinct reading comprehension datasets, WikiHop and MedHop [71]. WikiHop is open-domain and includes text from Wikipedia articles while MedHop comprised of paper abstracts sourced from the PubMed database. Both datasets require multiple inferences to be made by connecting facts from different documents.

RACE is a comprehensive reading comprehension dataset gathered from English exams meant for middle and high schoolers [72]. It features 28,000+ passages and almost 100,000 questions. The performance of models assessed by looking at their accuracy in middle school (RACE-m), high school (RACE-h) and overall, on the entire dataset (RACE). SQuAD is a unique dataset that comprises of questions asked by laypeople on Wikipedia articles, and the answers to those questions are selected segments of text from the related passage [73]. This dataset is gaining more attention among researchers due to its usefulness. Situations with adversarial generations (SWAG) is an expansive dataset for the challenge of grounded commonsense inference, which combines natural language inference and physically grounded thinking [74]. It comprises 113,000 multiple choice questions relating to ground-based situations. Large scale movie description challenge (LSMDC) and ActivityNet captions videos used to generate questions with four answer options, each indicating what might transpire next in the scene. To make sure machines do not get fooled, the actual video caption for the next event in the video is the correct answer. The other three options are incorrect ones generated by a computer and verified by humans, so they can trick machines but not people. RecipeQA is great dataset for understanding cooking recipes [75]. It features over 36,000 question-answer pairs developed from approximately 20,000 unique recipes with detailed instructions and images. The data can help improve the accuracy of multimodal comprehension of cooking recipes. RecipeQA solves the daunting task of understanding the multi-modal data comprising of images, titles and descriptions. To accurately provide answers to these questions, it requires sophisticated joint understanding of both image and text elements as well as temporal flow and procedural knowledge.

NarrativeQA offers a unique opportunity to gain better insights into natural language [76]. This dataset consists of 45,000 question-answer pairs related to full books and scripts, which encourages users to think critically when comprehending. This dataset consists of two components: i) comprehending summaries and ii) understanding full books or scripts. Both these features provide a helpful way to comprehend and interpret information better. DuoRC is a comprehensive collection of unique question-answer pairs generated from 7680 pairs of movie plots [77]. Each pair in the set presents two versions of the same movie, totaling 186,089 questions and answers. DuoRC is an exciting new natural language processing (NLP) development which encourages research in creating neural architectures that can stimulate knowledge and reasoning skills for reading comprehension issues. The Cosmos QA is a vast repository of 35,600 multiple-choice questions that demand commonsense-based reading comprehension [78]. This approach allows for a thoughtful analysis of everyday narratives from different points of view, asking questions that require reasoning beyond what explicitly stated in the text. It helps to gain a better understanding of the possible causes and outcomes based on the given context. Quasar is a dataset designed for open-domain question answering, which consists of two parts, Quasar-S and Quasar-T [79]. It has around 37,000 cloze-style queries created from definitions of software entity tags on Stack Overflow. Quasar-T is a collection of 43,000 open-domain trivia questions and their answers gathered from the web. SearchQA designed to be a comprehensive question-answer system featuring more than 140,000 question-answer pairs with an average of 49.6 snippets per pair [80]. Along with the question-answer tuples, it also contains meta-data, such as URLs of the respective snippets for each question-answer tuple. Ultimately, we opted for use data from SQuAD, Quasar, RACE, CoQA and MS MARCO. The final dataset contains approximately 300,000 records.

2.2. Training model

Neural question generation models have been split up into a range of categories, such as Seq2Seq models, pre-trained models, variational autoencoder models, graph-based models and adversarial network

models. The vast majority of modern NLP systems based on the Transformer architecture. Today there is a wide variety of different architectures. Of late, Transformer architecture [81] has demonstrated impressive capabilities for a variety of NLP tasks, managing to overcome structural issues caused by RNNs. Transformer technology utilizes a SeqSeq model to generate a symmetrical encoder and decoder, utilizing self-attention instead of requiring any recurrent gate. In order to adapt Transformer architectures for Seq2Seq tasks, Chan and Fan [82] proposed the using pre-trained bidirectional encoder representations from transformers (BERT) composed of transformers. They studied this in the context of question generation with answer span information. Wang *et al.* [83] suggested treating answer spans as the underlying basis for question generation and deploying transformer as the encoder and decoder module. Chai and Wan [24] proposed a semi-autoregressive approach to generate questions based on answer span data, with both the encoders and decoders taking the form of transformer architectures. The results of the study [84] showed that ChatGPT achieved a high accuracy rate of 87.5% in answering MCQs, with a mean response time of 3.5 seconds. The study also found that ChatGPT outperformed human experts in certain subjects, such as pharmacology and microbiology, while humans performed better in other subjects, such as pathology and clinical medicine [84]. Another works [85], [86] looked into fine tuning a pre-trained BART language model [87] to generate questions. This language model combines bidirectional and auto-regressive transformers for an improved performance. Wang *et al.* [86] appended an answer to its corresponding source article with a marker in between. It is noteworthy that References [85], [86] have utilized quality generators to evaluate the effectivity of abstractive summarization. This approach is new and engaging for question generation researchers and can open up interesting possibilities in the field.

Transformer architecture has been deployed in various works to address the task of answer agnostic question generation. Wang *et al.* [83] utilized the customary encoder-decoder architecture together with multi-head attention as a basic component. Kumar *et al.* [25] uncovered a powerful cross-lingual model to enhance the performance of the primary language's question generation (QG) by using resources from a secondary language. That accomplished through a Transformer-based encoder-decoder architecture. Scialom *et al.* [88] used transformers to add a copying mechanism, placeholders, and contextual word embeddings to the base QG architecture in order to create a system that is independent of the answers. Pan *et al.* [89] created a Chinese variety based question dataset from Baidu Zhidao by integrating context data and control signal to the transformer-based Seq2Seq model for generating unique questions through keywords. Laban *et al.* [90] modified a GPT2 language model [91] a transformer-based architecture for the QG task using the SQuAD 2.0 dataset as training data. Roemmele *et al.* [92] implemented a transformer-based Seq2Seq model with copying functions and devised various methods to supplement the training data. To improve the accuracy of MS MARCO, Nogueira *et al.* [93] used transformer-based Seq2Seq model T5 [94] to generate questions based on given passages. That helped to augment the original passages for better retrieval performance. Bhambhoria *et al.* [95] employed both the T5 transformer model and the rule-based method (a syntactic parser) to generate QA pairs for COVID-19 literature. In this study, we apply the BERT model and its detailed implementation. Our approach involves two main steps: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data by solving various pre-training problems. To perform fine-tuning, the BERT model is initialized with pre-trained parameters, after which all parameters are further tuned using task-specific labeled data. Each subsequent task includes individually tuned models, despite the fact that they are initialized with the same pre-trained parameters. The example of a question-answer system shown in Figure 1 serves as an illustrative example in the methods section.

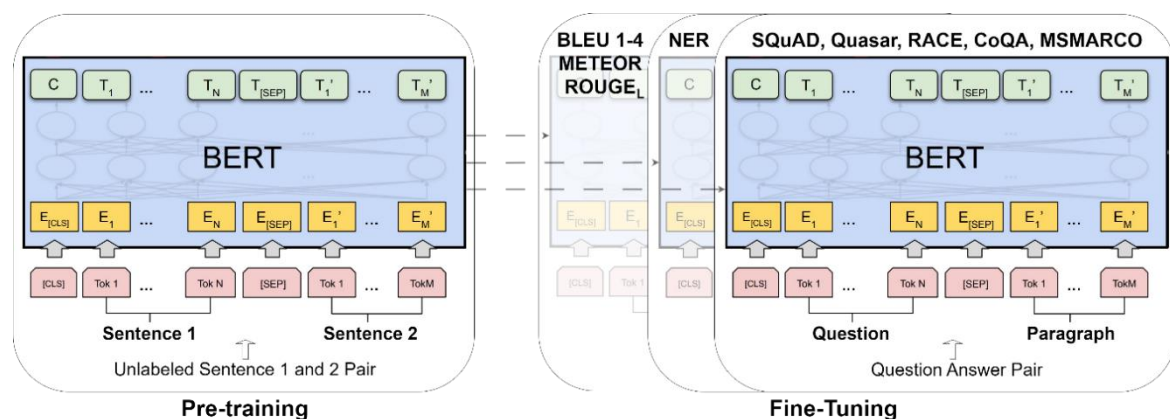


Figure 1. Overarching methodologies of the pre-training and fine-tuning processes for BERT

In the context of bidirectional encoder representations from transformers (BERT), the general processes involve pre-training and fine-tuning. The architectures remain consistent between these phases, excluding the output layers. The identical pre-trained model parameters serve to initialize models for diverse downstream tasks. In the fine-tuning stage, all parameters undergo refinement. Each input example is prefixed with the special symbol (CLS), while the special separator token (SEP) is employed to separate elements like questions and answers.

3. RESULTS AND DISCUSSION

3.1. Model training

After evaluating the different possible architectures, we settled on Google's T5 model [96]. The idea that forms the foundation of T5 is to convert all NLP tasks into sequential tasks. An AI model can be a great help to summarize or analyze text. When summarizing, it takes the text as input and produces the summary; for sentiment analysis, it also takes the analyzed text as input and provides a sequence indicating the sentiment of the text. Allowing a model to be repurposed for generating questions can be very useful since it was not written or pre-trained with that in mind. We need to feed the answer and context into the system, and it will give us the questions as results. The HuggingFace Transformers Python library [97] is a great tool that provides access to varied transformer models. By using this library, we can easily fetch pre-trained weights of T5 base model and use them for training question generation dataset. Loading the pre-trained model and tokenizer is a simple task. Once done, we can quickly encode the inputs, forward them into the model and produce an output. When we create a model to generate result, there must be a command to the model that any padding will be replaced by a value of -100. T5 ignores this part of the target when figuring out how well it is performing which makes it much more efficient. That must be done to prevent low loss values from being output, because any matching filling will be considered a correct prediction. We partitioned the training data into 85% for the training set and 15% for the validation set. We trained the model for 50 epochs on the dataset. The grammar in the output was correct.

3.2. Evaluation of generated questions

In order for the final system not to generate questions that are either not related to the answer or not related to the context, and also so that the resulting system does not generate some questions that were tautological or contained an answer within the question, it is necessary to train another model. This model should be able to evaluate and, in this way, filter the generated questions and answers. To complete this task, we opted for the pretrained version of BERT [98]. This transformer model trained on a cloze-style mechanism called masked language modeling, which basically fills in missing sections in sentences. Adopting this model as a pretraining objective has the significant benefit of requiring the model to understand text both before and after the gap in order to make accurate predictions. That creates bidirectional representations, which can be especially advantageous for certain types of tasks. BERT has revolutionized traditional language modeling goals. It enables the model to efficiently predict the next word in a sequence by understanding context from both directions. Thanks to BERT, tasks such as question and answer evaluation require less effort and provide better results when it comes to language understanding.

In order to perfect the model, we used the data from the question generator minus the context. During training, half of the cases will be given with a matching set of questions and answers while in other half they will be distorted. We have developed two mangling techniques to manipulate the answers: the first involves replacing it with an unrelated answer from the same set; and the second consists of taking the named entity from a question and inserting it into its response. The original aim of the study was to determine whether an answer was correct or not. Before any further training, the model based on pre-trained BERT achieved a 67% success rate on the validation set which is not much better than a throw of the dice. Training efforts paid off as we ultimately achieved a 93% accuracy rate enough to sift out the low-quality questions and answers.

We studied a system having two models: one that inputs answers and creates questions, and the other judging if the question-answer pairs are true or false. The overall system segments the text into sentences which serve as answers for further processing. The process of generating questions from the given answer options starts with combining them with text, encoding and passing it on to the question generation model. Subsequently, the inferred questions are combined with their respective answers and sent to the question-answer evaluation model for validating its accuracy. Our evaluator gives a score which helps indicate the accuracy of each question-answer pair. This score is used to rank them, and finally the N highest-ranking pairs are shown to the user.

3.3. Distractor generation

Multiple choice questions have added to this system, which can come in handy for creating quick assessments or simplifying the quiz process as students only need to pick a correct answer from the available set of options. Careless selection of alternative phrases may cause overly-simple questions that did not relate to the original inquiry. As a result, this approach may not yield substantial learning outcomes. A more holistic approach is needed to ensure students can gain adequate knowledge and have meaningful discussions. Adding an extra layer of complexity to the multiple-choice answers can be done using named entity recognition (NER). SpaCy offers this with in-built NER technology [99]. It involves extracting entities from the text, and then applying them as potential answers for our questions. For any given object type, alternative responses then chosen from the responses already provided.

As depicted in Figure 2, the process of question formation, evaluation, and distractor production divided into three steps. Step 1 (collecting the dataset) involves gathering pre-generated examples for teaching a neural network. These include sentences, sample questions, and the correct answers. Step 2 (generate QA pairs) in the process involves training a T5 model with the dataset to create question-answer pairs. Following this, pre-trained BERT is used at step 3 (assessing the adequacy of the generated pair) to evaluate the accuracy of these generated pairs. In step 4 (creating distractions) of the process, relevant distractions are created using a text passage and valid question/answer pair using the spaCy NER model.

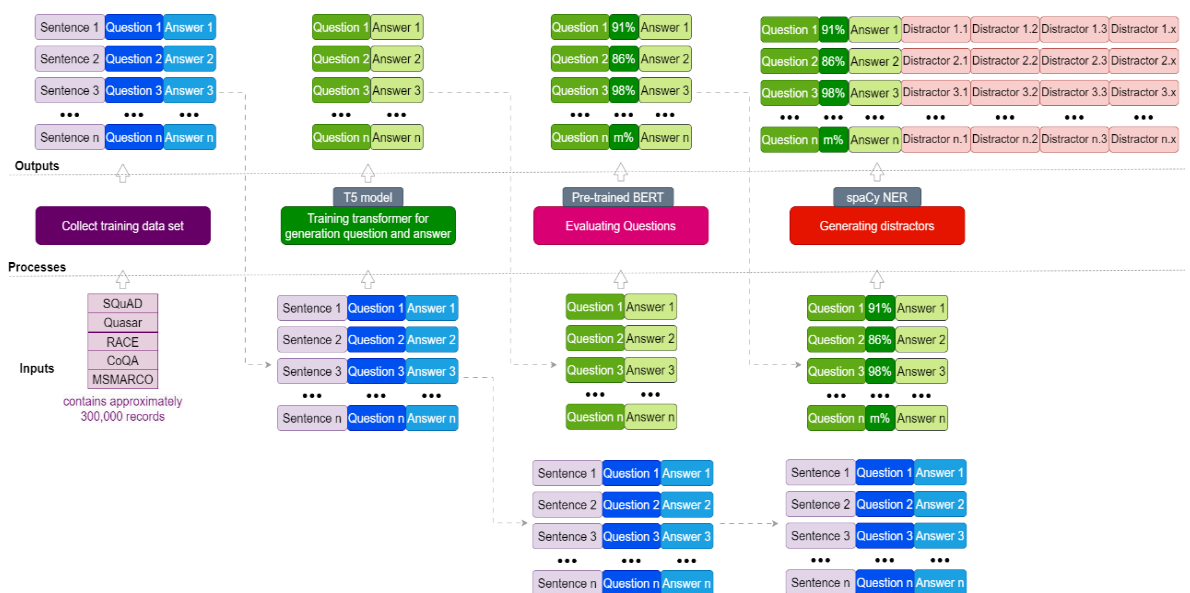


Figure 2. Pipeline for generating questionnaires composed of MCQs

3.4. Model evaluation

In order to demonstrate the effectiveness of our system, we compare it with a few other systems. We will summarize their strategies briefly, and describe the setup for running them, and evaluate their performance on our problem. The outcome of this comparison shown in Table 1. We adopt information retrieval (IR) baselines [55] to stop memorization of questions from the training set. To measure the gap between a question and an input sentence, two metrics employed: BM-25 [100] and edit distance [101]. By evaluating the set of metrics, the system is able to identify the most suitable question from the training set and assign it with a high scoring value.

SUM_{ROUGE} is a model and training procedure that produces successful results in text summarization on CNN/Daily Mail. It is particularly adept at dealing with longer output sequences [102]. The intra-attention decoder and combined training objectives applied to other sequence-to-sequence tasks that involve long inputs and outputs.

MOSES+ [103] is one of the most widely used statistical machine translation systems for sentence-to-question translations. It utilizes phrase-based language models to interpret source language text and generate questions in the target language. To bolster system performance, we trained a tri-gram language model on target side texts with the help of KenLM [104] and tuned it using minimum error rate training (MERT) on the development set. Performance results evaluated on the test set.

Seq2seq [55] is a sequence learning system for robotics and machine translation developed in Tensorflow. Before training or translating the inputted sequence reversed, and hyperparameters fine-tuned according to the development set. Finally, the model with best perplexity rate on the development set chosen.

M2S+cp is an efficient multi-perspective matching algorithm designed to generate questions automatically, thus helping to create a robust extractive QA system [23].

AutoQG_{QG+F+GAE} is a two-step approach designed to generate question-answer pairs from any text source. It helps to quickly create comprehensive QA, enabling a more thorough understanding of the topic at hand. This model combines a wide variety of approaches, like sequence-to-sequence models, Pointer Networks, entity alignment, and many more linguistic features. This way, it can identify useful responses from textual sources even for rare words. Furthermore, it can produce questions most related to the answer [105].

GE_{ROUGE+QSS+ANSS} is an AI-based approach towards developing an end-to-end solution for automatically generating questions using a generator-evaluator framework [106]. That enables a more comprehensive treatment of the entire question generation process. GE_{ROUGE+QSS+ANS} helps to take into account the syntax and semantics of questions, pinpoint critical answers, recognize words with contextual importance and omit any unimportant repeats. That also means that users can prioritize conformity with the structure of the original questions.

Table 1. Automated evaluation results of various BLEU 1–4, METEOR, and ROUGE_L question generation systems

| Model | BLEU 1 | BLEU 2 | BLEU 3 | BLEU 4 | METEOR | ROUGE _L |
|------------------------------------|--------|--------|--------|--------|--------|--------------------|
| IR _{BM25} | 5.18 | 0.91 | 0.28 | 0.12 | 4.57 | 9.16 |
| SUM _{ROUGE} | 11.94 | 3.95 | 1.65 | 0.082 | 6.61 | 16.17 |
| MOSES+ | 15.61 | 3.64 | 1.00 | 0.30 | 10.47 | 17.82 |
| IR _{Edit Distance} | 18.28 | 5.48 | 2.26 | 1.06 | 7.73 | 20.77 |
| seq2seq | 31.34 | 13.79 | 7.36 | 4.26 | 9.88 | 29.75 |
| M2S+cp | 32.04 | 21.72 | 15.87 | 13.98 | 18.77 | 32.71 |
| AutoQG _{QG+F+GAE} | 44.68 | 26.96 | 18.18 | 12.68 | 17.86 | 40.59 |
| GE _{ROUGE+QSS+ANSS} | 48.13 | 31.15 | 22.01 | 16.48 | 20.21 | 44.11 |
| Pre-trained _{T5+BERT+NER} | 52.58 | 36.27 | 25.15 | 17.59 | 28.03 | 49.66 |

3.5. Automatic evaluation

To assess our performance, we adopted the evaluation package provided by Chen *et al.* [107], which initially created to evaluate image captions. It involved BLEU 1, BLEU 2, BLEU 3, BLEU 4 [108], METEOR [109] and ROUGE_L [110] scripts. BLEU is a well-known metric that evaluates the average n-gram precision of a specific set of references sentences. It takes into account short sentences by providing an additional penalty. Additionally, BLEU score can further be improved by using up to n-grams for counting co-occurrences, labelled as BLEU-n scoring. METEOR is an effective metric that evaluates the similarity of a generated text to its reference by taking synonyms, stemming and paraphrasing into account. ROUGE utilized to assess the recall rate of summaries based on gold-standard sentences as a comparison. The results of the ROUGE_L (measured based on longest common subsequence) reported here.

3.6. Analysis of results

Table 1 indicates that the Pre-trained_{T5+BERT+NER} model, which only considers sentence-level info, performs better than all other models and baselines. The results of the automatic metric estimation are clear. It is evident that the IR, SUM, MOSES+ models are not up to the mark. That shows that simply memorizing the training set will not suffice for completing the task adequately. On the other hand, seq2seq and M2S+cp base models have earned good scores on BLEU 1 and ROUGE_L metrics which is quite reasonable considering sentence-question compatibilities. AutoQG and GE systems show similar performance levels, mostly making minor changes without actually paraphrasing. By looking at the results of our model, it becomes obvious that including pre-trained embeddings usually produces better results. Although the use of paragraph encoding may result in a slight drop in performance, it is still beneficial as it eliminates unnecessary noise while providing crucial useful information.

To assess the quality of our system, we conduct a qualitative assessment on the output samples and observe how closely they linked to the original input. In Table 2, we present sample questions generated by our model and GPT-3. In the first sentence, GPT-model made up a question to determine which of the scientists was the first to start research in the field of finite state machines. Whereas our model generated a question about what the first study about finite automata was. In the second, third and fourth sentences, a similar situation occurs, questions generated under different focuses. In the fifth sentence, the question is very similar. In terms of generating responses, GPT-model and our model give very similar results.

Additionally, the model-generated questions are more intricate which demonstrates that our system is highly effective.

Table 2. Comparison of questions and answers generated by our model and GPT-3 model

| Sentence | GPT-3 | Our model |
|--|--|--|
| In search of the simplest models to capture finite-state machines, Warren McCulloch and Walter Pitts were among the first researchers to introduce a concept similar to finite automata in 1943. | Question 1: Who were among the first researchers to introduce a concept similar to finite automata in 1943? Answer 1: Warren McCulloch and Walter Pitts. | Question 1: What was the first research on finite automata? Answer 1: In search of the simplest models to capture finite-state machines, Warren McCulloch and Walter Pitts were among the first researchers to introduce a concept similar to finite automata in 1943. |
| Overall, finite automata serve as powerful tools in computer science, enabling the modeling and analysis of systems with finite states, and contributing to the development of algorithms and techniques for solving a wide range of problems. | Question 2: What do finite automata enable in computer science? Answer 2: Finite automata enable the modeling and analysis of systems with finite states, and contribute to the development of algorithms and techniques for solving a wide range of problems. | Question 2: What is the main difference between finite automata and regular languages? Answer 2: Overall, finite automata serve as powerful tools in computer science, enabling the modeling and analysis of systems with finite states, and contributing to the development of algorithms and techniques for solving a wide range of problems. |
| A finite automata can be considered to be non-deterministic if it has more than one possible transition from one state on the same input symbol. | Question 3: When can a finite automata be considered non-deterministic? Answer 3: A finite automata can be considered non-deterministic if it has more than one possible transition from one state on the same input symbol. | Question 3: What is the definition of a finite automata? Answer 3: A finite automata can be considered to be non-deterministic if it has more than one possible transition from one state on the same input symbol. |
| Finite automata are widely used in computer science and mathematics for various applications, including designing and analyzing software and hardware systems, pattern recognition, and natural language processing. | Question 4: Where are finite automata widely used? Answer 4: Finite automata are widely used in computer science and mathematics for various applications, including designing and analyzing software and hardware systems, pattern recognition, and natural language processing. | Question 4: What is the definition of finite automata? Answer 4: Finite automata are widely used in computer science and mathematics for various applications, including designing and analyzing software and hardware systems, pattern recognition, and natural language processing. |
| Finite automata, also known as finite state machines, are computational models that represent systems with a finite number of states and transitions between those states. | Question 5: What are finite automata also known as? Answer 5: Finite automata are also known as finite state machines. | Question 5: What are finite automata? Answer 5: Finite state machines. |

4. CONCLUSION

The article describes a comparison between the pre-trained $T5_{+BERT+NER}$ system and other systems. Research results shows that the pre-trained $T5_{+BERT+NER}$ model, which considers sentence-level information, outperforms all other models and baselines. Some models, such as IR, SUM, and MOSES+, did not meet expectations, indicating that simply memorizing the training set is not sufficient. Seq2seq and M2S+cp base models performed well on certain metrics, considering sentence-question compatibility. AutoQG and GE systems had similar performance levels but made minor changes without truly paraphrasing. Our model, which includes pre-trained embeddings, consistently produced better results. Although paragraph encoding slightly decreased performance, it removed unnecessary noise while providing important information. A qualitative assessment was conducted by comparing sample questions generated by our model and GPT-3. The Pre-trained $T5_{+BERT+NER}$ model generated more relevant questions with intricate details, demonstrating its effectiveness.

ACKNOWLEDGEMENTS

This research is funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. AP14869848).

REFERENCES

- [1] M. Heilman and N. A. Smith, "Good question! Statistical ranking for question generation," *NAACL HLT 2010 - Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference*, pp. 609–617, 2010, doi: 10.5555/1857999.1858085.

- [2] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A systematic review of automatic question generation for educational purposes," *International Journal of Artificial Intelligence in Education*, vol. 30, no. 1, pp. 121–204, Nov. 2020, doi: 10.1007/s40593-019-00186-y.
- [3] R. Zhang, J. Guo, L. Chen, Y. Fan, and X. Cheng, "A review on question generation from natural language text," *ACM Transactions on Information Systems*, vol. 40, no. 1, pp. 1–43, Sep. 2022, doi: 10.1145/3468889.
- [4] D. R. Ch and S. K. Saha, "Automatic multiple choice question generation from text: A survey," *IEEE Transactions on Learning Technologies*, vol. 13, no. 1, pp. 14–25, Jan. 2020, doi: 10.1109/TLT.2018.2889100.
- [5] O. Rodríguez Rocha and C. Faron Zucker, "Automatic generation of quizzes from DBpedia according to educational standards," in *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*, 2018, pp. 1035–1041, doi: 10.1145/3184558.3191534.
- [6] M. Divate and A. Salgaonkar, "Automatic question generation approaches and evaluation techniques," *Current Science*, vol. 113, no. 9, pp. 1683–1691, Nov. 2017, doi: 10.18520/cs/v113/i09/1683-1691.
- [7] B. Das, M. Majumder, S. Phadikar, and A. A. Sekh, "Multiple-choice question generation with auto-generated distractors for computer-assisted educational assessment," *Multimedia Tools and Applications*, vol. 80, no. 21–23, pp. 31907–31925, Sep. 2021, doi: 10.1007/s11042-021-11222-2.
- [8] J. Mostow and W. Chen, "Generating instruction automatically for the reading strategy of self-questioning," *Frontiers in Artificial Intelligence and Applications*, vol. 200, no. 1, pp. 465–472, 2009, doi: 10.3233/978-1-60750-028-5-465.
- [9] A. Varga and L. A. Ha, "Wlv: A question generation system for the QGSTEC 2010 Task B," in *Proceedings of QG2010: The third workshop on question generation*, 2010, pp. 80–83.
- [10] S. Kalady, A. Elikkottil, and R. Das, "Natural language question generation using syntax and keywords," in *Proceedings of QG2010: The Third Workshop on Question Generation*, 2010, vol. 2, pp. 5-14.
- [11] H. Ali, Y. Chali, and S. a. Hasan, "Automatic question generation from sentences," *Proceedings of TALN 2010*, 2010, pp. 19–23.
- [12] P. Mannem, R. Prasad, and A. Joshi, "Question generation from paragraphs at UPenn: QGSTEC system description," in *Proceedings of QG 2010: The Third Workshop on Question Generation, Pittsburg, PA*, Aug. 2010, pp. 84–91.
- [13] Y. Huang and L. He, "Automatic generation of short answer questions for reading comprehension assessment," *Natural Language Engineering*, vol. 22, no. 3, pp. 457–489, Jan. 2016, doi: 10.1017/S1351324915000455.
- [14] X. Yao and Y. Zhang, "Question generation with minimal recursion semantics," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010, pp. 1067–1076.
- [15] A. Copestake, D. Flickinger, C. Pollard, and I. A. Sag, "Minimal recursion semantics: An introduction," *Research on Language and Computation*, vol. 3, no. 2, pp. 281–332, Jul. 2005, doi: 10.1007/s11168-006-6327-9.
- [16] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017, vol. 1, pp. 1342–1352, doi: 10.18653/v1/P17-1123.
- [17] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2015.
- [18] B. Das, M. Majumder, S. Phadikar, and A. A. Sekh, "Automatic question generation and answer assessment: a survey," *Research and Practice in Technology Enhanced Learning*, vol. 16, no. 1, Mar. 2021, doi: 10.1186/s41039-021-00151-1.
- [19] K. Mazidi and P. Tarau, "Infusing NLU into automatic question generation," in *INLG 2016 - 9th International Natural Language Generation Conference, Proceedings of the Conference*, 2016, pp. 51–60, doi: 10.18653/v1/w16-6609.
- [20] B. Liu *et al.*, "Learning to generate questions by learning what not to generate," in *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, May 2019, pp. 1106–1118, doi: 10.1145/3308558.3313737.
- [21] W. Zhou, M. Zhang, and Y. Wu, "Question-type driven question generation," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 6032–6037, doi: 10.18653/v1/d19-1622.
- [22] W. Hu, B. Liu, J. Ma, D. Zhao, and R. Yan, "Aspect-based question generation," in *6th International Conference on Learning Representations*, pp. 1-10, 2018.
- [23] L. Song, Z. Wang, W. Hamza, Y. Zhang, and D. Gildea, "Leveraging context information for natural question generation," in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2018, vol. 2, pp. 569–574, doi: 10.18653/v1/n18-2090.
- [24] Z. Chai and X. Wan, "Learning to ask more: Semi-autoregressive sequential question generation under dual-graph interaction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 225–237, doi: 10.18653/v1/2020.acl-main.21.
- [25] V. Kumar, N. Joshi, A. Mukherjee, G. Ramakrishnan, and P. Jyothi, "Cross-lingual training for automatic question generation," in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020, pp. 4863–4872, doi: 10.18653/v1/p19-1481.
- [26] L. Dong *et al.*, "Unified language model pre-training for natural language understanding and generation," *Advances in Neural Information Processing Systems*, vol. 32, pp. 1-15, 2019, doi: 10.5555/3294996.3295135.
- [27] W. Wang, S. Feng, D. Wang, and Y. Zhang, "Answer-guided and semantic coherent question generation in open-domain conversation," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 5066–5076, doi: 10.18653/v1/d19-1511.
- [28] Y. Chen, L. Wu, and M. J. Zaki, "Reinforcement learning based graph-to-sequence model for natural question generation," in *8th International Conference on Learning Representations*, 2019.
- [29] J. Bao, Y. Gong, N. Duan, M. Zhou, and T. Zhao, "Question generation with doubly adversarial nets," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 11, pp. 2230–2239, Nov. 2018, doi: 10.1109/TASLP.2018.2859777.
- [30] T. Wang, X. Yuan, and A. Trischler, "A joint model for question answering and question generation," *arxiv.org/abs/1706.01450*, Jun. 2017.
- [31] Z. Fan, Z. Wei, S. Wang, Y. Liu, and X. Huang, "A reinforcement learning framework for natural question generation using bi-discriminators," in *COLING 2018 - 27th International Conference on Computational Linguistics, Proceedings*, Aug. 2018, pp. 1763–1774.
- [32] Y. H. Liao and J. L. Koh, "Question generation through transfer learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12144 LNAI, Springer International Publishing, 2020, pp. 3–17, doi: 10.1007/978-3-030-55789-8_1.




- [33] S. Rakangor and Y. R. Ghodasara, "Literature review of automatic question generation systems," in *International Journal of Scientific and Research Publications*, 2015, vol. 5, no. 1, pp. 2250–3153.
- [34] A. Santhanavijayan *et al.*, "Automatic generation of multiple choice questions for e-assessment," *International Journal of Signal and Imaging Systems Engineering*, vol. 10, no. 1/2, p. 54, 2017, doi: 10.1504/IJSISE.2017.084571.
- [35] R. Patra and S. K. Saha, "A hybrid approach for automatic generation of named entity distractors for multiple choice questions," *Education and Information Technologies*, vol. 24, no. 2, pp. 973–993, Sep. 2019, doi: 10.1007/s10639-018-9814-3.
- [36] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014, vol. 2014-June, pp. 55–60, doi: 10.3115/v1/p14-5010.
- [37] T. Effenberger, "Automatic question generation and adaptive practice," Doctoral dissertation, Fakultä informatiky, Masarykova univerzita, 2015.
- [38] S. Smith, P. V. S. Avinesh, and A. Kilgarriff, "Gap-fill tests for language learners: Corpus-driven item generation," in *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*, pp. 1-6, 2010.
- [39] B. Das and M. Majumder, "Factual open cloze question generation for assessment of learner's knowledge," *International Journal of Education and Information Technologies*, vol. 14, no. 1, Aug. 2017, doi: 10.1186/s41239-017-0060-3.
- [40] L. Becker, S. Basu, and L. Vanderwende, "Mind the gap: learning to choose gaps for question generation," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 742-751, 2012.
- [41] D. Coniam, "A preliminary inquiry into using corpus word frequency data in the automatic generation of English language cloze tests," *CALICO Journal*, vol. 14, no. 2–4, pp. 15–33, Jan. 2013, doi: 10.1558/cj.v14i2-4.15-33.
- [42] N. Karamanis, L. A. Ha, and R. Mitkov, "Generating multiple-choice test items from medical text," in *Proceedings of the Fourth International Natural Language Generation Conference on - INLG '06*, 2006, p. 111, doi: 10.3115/1706269.1706291.
- [43] R. Mitkov, L. A. Ha, and N. Karamanis, "A computer-aided environment for generating multiple-choice test items," *Natural Language Engineering*, vol. 12, no. 2, pp. 177–194, May 2006, doi: 10.1017/S1351324906004177.
- [44] D. Gates, G. Aist, J. Mostow, M. Mckeown, and J. Bey, "How to generate cloze questions from definitions: a syntactic approach," in *2011 AAAI Fall Symposium Series*, 2011.
- [45] N. Afzal and R. Mitkov, "Automatic generation of multiple choice questions using dependency-based semantic relations," *Soft Computing*, vol. 18, no. 7, pp. 1269–1281, Oct. 2014, doi: 10.1007/s00500-013-1141-4.
- [46] M. Majumder and S. K. Saha, "Automatic selection of informative sentences: The sentences that can generate multiple choice questions," *Knowledge Management and E-Learning*, vol. 6, no. 4, pp. 377–391, Dec. 2014, doi: 10.34105/j.kmel.2014.06.025.
- [47] M. Agarwal, R. Shah, and P. Mannem, "Automatic question generation using discourse cues," in *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, 2011, pp. 1-9.
- [48] K. Mazidi and R. D. Nielsen, "Linguistic considerations in automatic question generation," in *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 2014, vol. 2, pp. 321–326, doi: 10.3115/v1/p14-2053.
- [49] L. Gao, K. Gimpel, and A. Jensson, "Distractor analysis and selection for multiple-choice cloze questions for second-language learners," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 102–114, doi: 10.18653/v1/2020.bea-1.10.
- [50] M. Agarwal and P. Mannem, "Automatic gap-fill question generation from textbooks," in *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, 2011, pp. 56-64.
- [51] S. Knoop and S. Wilske, "Wordgap-automatic generation of gap-filling vocabulary exercises for mobile learning," in *Proceedings of the Second Workshop on NLP for Computer-Assisted Language Learning at NODALIDA 2013*, 2013, pp. 39-47.
- [52] J. Leo *et al.*, "Ontology-based generation of medical, multi-term MCQs," *International Journal of Artificial Intelligence in Education*, vol. 29, no. 2, pp. 145–188, Jan. 2019, doi: 10.1007/s40593-018-00172-w.
- [53] I. Aldabe and M. Maritxalar, "Automatic distractor generation for domain specific texts," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6233 LNAI, Springer Berlin Heidelberg, 2010, pp. 27–38, doi: 10.1007/978-3-642-14770-8_5.
- [54] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arxiv.org/abs/1301.3781*, Jan. 2013.
- [55] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [56] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for sentence summarization," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 379–389, doi: 10.18653/v1/d15-1044.
- [57] S. Iyer, I. Konstas, A. Cheung, and L. Zettlemoyer, "Summarizing source code using a neural attention model," in *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2016, vol. 4, pp. 2073–2083, doi: 10.18653/v1/p16-1195.
- [58] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," *Proceedings of the 32nd International Conference on Machine Learning, PMLR*, pp. 2048–2057, Jun. 01, 2015. Accessed: Dec. 12, 2023. [Online]. Available: <https://proceedings.mlr.press/v37/xuc15.html>
- [59] T. Tãm, N. C. Ú U. Vã, C. Ê N. Giao, C. Ngh, and Å N B Û I Chu, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North*, 2016, vol. 01, pp. 1–23, doi: 10.18653/v1/N19-1423.
- [60] D. Bakır and M. S. Aktas, "A systematic literature review of question answering: research trends, datasets, methods," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13377 LNCS, Springer International Publishing, 2022, pp. 47–62, doi: 10.1007/978-3-031-10536-4_4.
- [61] M. Boratko *et al.*, "A systematic classification of knowledge, reasoning, and context within the ARC dataset," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 60–70, doi: 10.18653/v1/w18-2607.
- [62] M. Saeidi *et al.*, "Interpretation of natural language rules in conversational machine reading," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018, pp. 2087–2097, doi: 10.18653/v1/d18-1233.
- [63] S. Soni and K. Roberts, "Evaluation of dataset selection for pre-training and fine-tuning transformer language models for clinical question answering," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 5532-5538.
- [64] M. Gambhir and V. Gupta, "Deep learning-based extractive text summarization with word-level attention mechanism," *Multimedia Tools and Applications*, vol. 81, no. 15, pp. 20829–20852, Mar. 2022, doi: 10.1007/s11042-022-12729-y.

- [65] S. Reddy, D. Chen, and C. D. Manning, "CoQA: A conversational question answering challenge," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 249–266, Nov. 2019, doi: 10.1162/tacl_a_00266.
- [66] T. Wolfson *et al.*, "Break it down: A question understanding benchmark," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 183–198, Dec. 2020, doi: 10.1162/tacl_a_00309.
- [67] P. Bajaj *et al.*, "MS MARCO: A human generated machine reading comprehension dataset," *CEUR Workshop Proceedings*, Nov. 2016.
- [68] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth, "Looking beyond the surface: A challenge set for reading comprehension over multiple sentences," in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2018, vol. 1, pp. 252–262, doi: 10.18653/v1/n18-1023.
- [69] T. Kwiatkowski *et al.*, "Natural questions: A benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, Nov. 2019, doi: 10.1162/tacl_a_00276.
- [70] I. Maqsood, "Evaluating newsQA dataset with ALBERT," in *2022 17th International Conference on Emerging Technologies, ICET 2022*, Nov. 2022, pp. 1–5, doi: 10.1109/ICET56601.2022.10004665.
- [71] Y. Cao, M. Fang, and D. Tao, "BAG: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering," in *Proceedings of the 2019 Conference of the North*, 2019, pp. 357–362, doi: 10.18653/v1/N19-1032.
- [72] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale ReAding comprehension dataset from examinations," in *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017, pp. 785–794, doi: 10.18653/v1/d17-1082.
- [73] T. Parshakova, F. Rameau, A. Serdega, I. S. Kweon, and D. S. Kim, "Latent question interpretation through variational adaptation," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 27, no. 11, pp. 1713–1724, Nov. 2019, doi: 10.1109/TASLP.2019.2929647.
- [74] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "SWAG: A large-scale adversarial dataset for grounded commonsense inference," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018, pp. 93–104, doi: 10.18653/v1/d18-1009.
- [75] S. Yagcioglu, A. Erdem, E. Erdem, and N. Iklizler-Cinbis, "RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018, pp. 1358–1368, doi: 10.18653/v1/d18-1166.
- [76] S. Indurthi, S. Yu, S. Back, and H. Cuayáhuitl, "Cut to the chase: A context zoom-in network for reading comprehension," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018, pp. 570–575, doi: 10.18653/v1/d18-1054.
- [77] A. Saha, R. Aralikkatte, M. M. Khapra, and K. Sankaranarayanan, "DuoRC: Towards complex language understanding with paraphrased reading comprehension," in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018, vol. 1, pp. 1683–1693, doi: 10.18653/v1/p18-1156.
- [78] L. Huang, R. Le Bras, C. Bhagavatula, and Y. Choi, "COSMOS QA: Machine reading comprehension with contextual commonsense reasoning," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 2391–2401, doi: 10.18653/v1/d19-1243.
- [79] B. Dhingra, K. Mazaitis, and W. W. Cohen, "Quasar: Datasets for question answering by search and reading," *arxiv.org/abs/1707.03904*, Jul. 2017.
- [80] K. Lee, K. Yoon, S. Park, and S. Hwang, "Semi-supervised training data generation for multilingual question answering," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 2018.
- [81] A. Vaswani *et al.*, "Attention is all you need," *arxiv.org/abs/1706.03762*, Jun. 2017.
- [82] Y. H. Chan and Y. C. Fan, "A recurrent bert-based model for question generation," in *MRQA@EMNLP 2019 - Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019, pp. 154–162, doi: 10.18653/v1/d19-5821.
- [83] X. Wang, B. Wang, T. Yao, Q. Zhang, and J. Xu, "Neural question generation with answer pivot," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 9138–9145, Apr. 2020, doi: 10.1609/aaai.v34i05.6449.
- [84] S. A. Meo, A. A. Al-Masri, M. Alotaibi, M. Z. S. Meo, and M. O. S. Meo, "ChatGPT knowledge evaluation in basic and clinical medical sciences: multiple choice question examination-based performance," *Healthcare (Switzerland)*, vol. 11, no. 14, Jul. 2023, doi: 10.3390/healthcare11142046.
- [85] E. Durmus, H. He, and M. Diab, "FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5055–5070, doi: 10.18653/v1/2020.acl-main.454.
- [86] A. Wang, K. Cho, and M. Lewis, "Asking and answering questions to evaluate the factual consistency of summaries," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5008–5020, doi: 10.18653/v1/2020.acl-main.450.
- [87] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.
- [88] T. Scialom, B. Piwowarski, and J. Staiano, "Self-attention architectures for answer-agnostic neural question generation," in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020, pp. 6027–6032, doi: 10.18653/v1/p19-1604.
- [89] Y. Pan, B. Hu, Q. Chen, Y. Xiang, and X. Wang, "Learning to generate diverse questions from keywords," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, May 2020, vol. 2020-May, pp. 8224–8228, doi: 10.1109/ICASSP40776.2020.9053822.
- [90] P. Laban, J. Canny, and M. A. Hearst, "What's the latest? A question-driven news chatbot," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 380–387, doi: 10.18653/v1/2020.acl-demos.43.
- [91] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, 2019.
- [92] M. Roemmele, D. Sidhpura, S. DeNeefe, and L. Tsou, "Answer quest: A system for generating question-answer items from multi-paragraph documents," in *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the System Demonstrations*, 2021, pp. 40–52, doi: 10.18653/v1/2021.eacl-demos.6.
- [93] R. Nogueira and J. Lin, "From doc2query to docTTTTTquery An MS MARCO passage retrieval task [1] μpublication", Accessed: Dec. 12, 2023. [Online]. Available: <https://github.com/castorini/docTTTTTquery>




- [94] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *arxiv.org/abs/1910.10683*, Oct. 2019.
- [95] R. Bhambhoria *et al.*, “A smart system to generate and validate question answer pairs for COVID-19 literature,” in *Proceedings of the First Workshop on Scholarly Document Processing*, 2020, pp. 20–30, doi: 10.18653/v1/2020.sdp-1.4.
- [96] L. Qin, A. Gupta, S. Upadhyay, L. He, Y. Choi, and M. Faruqui, “TIMEDIAL: Temporal commonsense reasoning in dialog,” in *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2021, pp. 7066–7076, doi: 10.18653/v1/2021.acl-long.549.
- [97] L. Yabloko, “ETHAN at SemEval-2020 Task 5: Modelling Causal Reasoning in Language using neuro-symbolic cloud computing,” in *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, 2020, pp. 645–652, doi: 10.18653/v1/2020.semeval-1.83.
- [98] C. K. Lo, “YiSi - A unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources,” in *WMT 2019 - 4th Conference on Machine Translation, Proceedings of the Conference*, 2019, vol. 2, pp. 507–513, doi: 10.18653/v1/w19-5358.
- [99] B. Kleinberg, M. Mozes, A. Arntz, and B. Verschuere, “Using named entities for computer-automated verbal deception detection,” *Journal of Forensic Sciences*, vol. 63, no. 3, pp. 714–723, Sep. 2018, doi: 10.1111/1556-4029.13645.
- [100] S. E. Robertson and S. Walker, “Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval,” in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, Springer London, 1994, pp. 232–241, doi: 10.1007/978-1-4471-2099-5_24.
- [101] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet physics. Doklady*, vol. 10, pp. 707–710, 1965.
- [102] R. Paulus, C. Xiong, and R. Socher, “A deep reinforced model for abstractive summarization,” in *6th International Conference on Learning Representations*, 2017.
- [103] P. Koehn *et al.*, “Moses: Open source toolkit for statistical machine translation,” in *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 2007, pp. 177–180.
- [104] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, “Scalable modified kneser-ney language model estimation,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, vol. 2, 2013, pp. 690–696.
- [105] V. Kumar, K. Boorla, Y. Meena, G. Ramakrishnan, and Y. F. Li, “Automating reading comprehension by generating question and answer pairs,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10939 LNAI, Springer International Publishing, 2018, pp. 335–348, doi: 10.1007/978-3-319-93040-4_27.
- [106] V. Kumar, G. Ramakrishnan, and Y. F. Li, “Putting the horse before the cart: A generator-evaluator framework for question generation from text,” in *CoNLL 2019 - 23rd Conference on Computational Natural Language Learning, Proceedings of the Conference*, 2019, pp. 812–821, doi: 10.18653/v1/k19-1076.
- [107] X. Chen *et al.*, “Microsoft COCO captions: Data collection and evaluation server,” *arxiv.org/abs/1504.00325v2*, Apr. 2015.
- [108] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002, vol. 2002-July, pp. 311–318, doi: 10.3115/1073083.1073135.
- [109] M. Denkowski and A. Lavie, “Meteor universal: Language specific translation evaluation for any target language,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 376–380, doi: 10.3115/v1/w14-3348.
- [110] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text summarization branches out*, pp. 74–81, 2004.

BIOGRAPHIES OF AUTHORS



Alibek Barlybayev    received the B.Eng. degree in information systems from L.N. Gumilyov Eurasian National University, Kazakhstan, in 2009 and the M.S. and Ph.D. degrees in computer science from L.N. Gumilyov Eurasian National University, Kazakhstan, in 2011 and 2015, respectively. Currently, he is a Director of the Research Institute of Artificial Intelligence, L.N. Gumilyov Eurasian National University. He is also an associate professor of the Department of Artificial Intelligence Technologies, L.N. Gumilyov Eurasian National University, and Higher School of Information Technology and Engineering, Astana International University. His research interests are NLP, the use of neural networks in word processing, smart textbooks, fuzzy logic, stock market price prediction, information security. He can be contacted at email: frank-ab@mail.ru.



Bakhyt Matkarimov    holds a master's degree from Novosibirsk State University, Russia. Also, he holds a PhD, Institute of Mathematics, Almaty, Kazakhstan, and Dr.Sci., Institute of Mathematics, Almaty, Kazakhstan. Awards: In 2022, he was honored as the Best Researcher of the Republic of Kazakhstan, a testament to the impact of his work on the scientific landscape. His commitment to the development of science was acknowledged in 2016 when he received an award for merits in the field from the Republic of Kazakhstan. In 2008, he was bestowed the title of Honorary Worker of Education of the Republic of Kazakhstan. In 1997, he was granted a scholarship by the Swiss Academy of Engineering Sciences in Switzerland. He is currently a research lecturer of the Department of Artificial Intelligence Technologies, L.N. Gumilyov Eurasian National University. His scientific interests are bioinformatics, computer vision, neural networks, question-answer systems. He can be contacted at email: bakhyt.matkarimov@gmail.com.