GENERATING THE QUESTION FROM THE ANSWER

Talha Çelik, Hüseyin Said Arıcı
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, 34220 Istanbul, Türkiye
{talha.celik1, said.arici}@std.yildiz.edu.tr

***Özetçe* —Bu çalışmada, verilen bir cevaptan soruyu üretmek amacıyla bir doğal dil işleme modeli geliştirilmiştir. Bu doğrultuda, modelin giriş verisi olarak metnin tokenlerinin tersine çevrilmiş hali kullanılarak dil modelinin eğitimi gerçekleştirilmiştir. Tersine çevrilmiş token dizileriyle çalışmanın temel amacı, baz alınan gpt modelinde kullanılan önceki token tahmin sistemidir. Eğitim sürecinde farklı öğrenme oranları kullanılarak farklı modeller eğitilmiştir. Eğitim boyunca her iki modelin eğitim ve doğrulama kayıpları takip edilerek performansları değerlendirilmiştir. Model performansını değerlendirmek için metrikler kullanılmıştır. Modelden çıktı alma noktasında farklı parametre seçenekleri sunulmuştur. Bu sayede daha yaratıcı cevaplara ulaşılabilmesi sağlanmıştır. Bu modeller, bir yerel arayüz aracılığıyla kullanıma sunulmuştur.**

Anahtar Kelimeler—doğal dil işleme, yapay zeka, sorudan cevap üretimi, dil modelleri.

***Abstract*—In this study, a natural language processing (NLP) model was developed with the aim of generating a question based on a given answer. Accordingly, the model was trained using reversed token sequences as input. The primary purpose of working with reversed token sequences is to adopt the previous token prediction system used in the baseline GPT model. Different models were trained using various learning rates during the training process. Throughout the training, the training and validation losses of both models were tracked, and their performance was evaluated. Metrics were used to assess model performance. Additionally, various parameter options were provided for generating outputs from the model, allowing for more creative responses. These models were made available through a local interface for user interaction.**

Keywords—natural language processing, artificial intelligence, question generation, language models.

I. Introduction

In recent years, text generation-focused artificial intelligence technologies have become a significant focal point to meet the rapidly evolving needs of the digital world and to make access to information faster and more efficient. Artificial intelligence (AI), especially with advancements in fields such as NLP and machine learning, has revolutionized text creation processes and demonstrated its potential in this area. These systems, which can now not only understand texts but also analyze complex language structures and contextual relationships to generate original content, have significantly reduced the workload of both individuals and organizations in content production.

AI-powered text generation technologies find a broad range of applications, from text-based data analysis to customer services, marketing, and education. These technologies provide a substantial advantage by enabling the production of an almost limitless variety of content, particularly on platforms where content is rapidly consumed, such as social media, digital marketing, news sites, and academic institutions. For example, according to the e-learning industry, forty-seven percent of learning management tools will be driven by AI capabilities within the next three years.[1]

A. Objective

The objective of this project is to enable a model to generate a question based on a given answer by utilizing current natural language processing techniques. This model will apply a reverse approach of a GPT-like model, generating a question from an answer and thus introducing a novel approach to AI-driven question generation. Throughout the project, efforts will focus on improving the model's ability to accurately predict the correct question using training data. Additionally, the effect of the fine-tuning process after training will be observed and analyzed.

B. Preliminary Review

As a result of the preliminary review, it was observed that a similar study was submitted to TÜBİTAK as a research paper in 2022 [2]. In that study, the mT5 model, a multilingual language model, was utilized. mT5 is based on an encoder-decoder architecture, providing flexibility in various text processing tasks. Since it was previously trained on different languages, it is also suitable for use in Turkish. By retraining the model with Turkish datasets, it was enabled to generate questions and extract answers from Turkish texts. The model was trained in a multi-task manner to perform both question generation and answer extraction simultaneously. This approach enables the development of a fully automated system for question generation without requiring any manual intervention during the process.

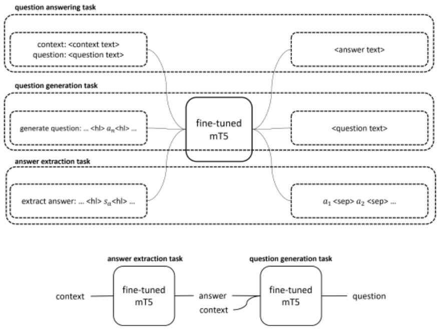

Figure 1 Structure of mT5

II. Literature Review

A notable study by Katira Soleymanzadeh at Ege University focused on generating automatic questions from Turkish texts in the field of biology. The process involved analyzing meaningful sentences to create question sentences. Dependency parsing was employed to identify relationships between words and sentence elements, while semantic analysis was conducted using a domain-specific proposition bank and a semantic role labeling system. Questions such as “who,” “what,” “where,” “how,” and “why” were generated by identifying and labeling semantic roles.

The system operates in three main stages:

Semantic Role Labeling Dataset Creation: A dataset was created by selecting and labeling key biological terms and concepts in sentences.

Sentence Structure Extraction: Dependency parsing results were combined with semantic role labels to extract sentence structures.

Question Generation: Appropriate question types were selected, and questions were generated using predefined templates.

This automatic question generation system was designed to aid teachers in evaluating students’ knowledge and to facilitate self-study and self-assessment for students. As one of the first studies to attempt automatic question generation (AQG) in Turkish texts using semantic analysis, the study demonstrates that the method can be adapted for other fields.[3]

Another significant study by Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung from Seoul National University aimed to improve neural question generation (NQG) performance by better utilizing the target answer. Existing NQG models often produced undesirable results by directly including target words in the questions. To address this issue, a model based on an “answer separation” approach was developed. By replacing the target answer with a special token, the model was able to better understand the context and determine the appropriate question type.

Additionally, a “key attention” module was introduced to help the model focus on the key information within the target answer. Experimental results on the SQuAD dataset showed that the proposed model outperformed existing NQG models, generating more accurate questions with fewer instances of target answer inclusion.[4]

Xingwu Sun and colleagues tackled two main issues in NQG models: mismatched question types and the inclusion of irrelevant words in generated questions. They proposed a model with “answer-focused” and “position-aware” mechanisms. The answer-focused mechanism ensured that generated question words aligned with the answer type, while the position-aware mechanism prevented irrelevant words from being included by focusing on the context surrounding the answer.

Their model was tested on the SQuAD and MARCO datasets, achieving superior performance compared to previous methods. The experimental results showed a reduction in error rates, improved question word selection, and more accurate outcomes.[5]

Another study by Kettip Kriangchaivech and Artit Wangperawong focused on automatic question generation from Wikipedia texts using a transformer-based model. The model was trained on the SQuAD dataset and demonstrated the capability to generate questions from new texts. Compared to older RNN-based models, transformer-based models offer faster training and better performance.

This approach is anticipated to simplify the process of creating exam and test questions for educators. The model generates questions based on context and answers, with word error rate metrics used to compare the generated questions with the original ones. The results indicated that the transformer-based model produces syntactically correct and contextually relevant questions.[6]

Ying-Hong Chan and Yao-Chung Fan explored the use of BERT-based models for question generation tasks and aimed to enhance performance by developing new architectures. Initially, the direct use of BERT was found inadequate due to the lack of sequential information. Consequently, they developed the BERT-SQG model, which incorporates outputs from previous steps.

Additionally, the BERT-HLSQG model was designed by introducing special marking tokens to reduce answer ambiguities and further improve performance. Tested on the SQuAD dataset, these models achieved state-of-the-art results, raising the BLEU 4 score from 16.85 to 22.17. The models demonstrated high performance even in paragraph-level contexts, producing questions of near-human quality.[7]

III. System Design and Methods

The system fine-tunes a base model using a prepared question-answer dataset on Google Colab, saves the model, and makes it ready for use. During the preprocessing steps, the available dataset is transformed into a format suitable for training. To optimize the GPT model, BitsAndBytes is utilized to load the model in 8-bit precision. Several methods were applied to ensure the proper functioning of the model. The first method involved reversing the tokens in the dataset to enable the model to predict previous tokens accurately. This approach ensures that while predicting the next token, the model effectively works backward. Additionally, to prevent the model from continuing beyond the desired point after identifying the target question, the "endoftext" token was inserted at appropriate parts of the data. The dataset generated using artificial intelligence was utilized for training the model with PyTorch. The fine-tuned instruct model was then saved on Google Drive and is ready for reuse.

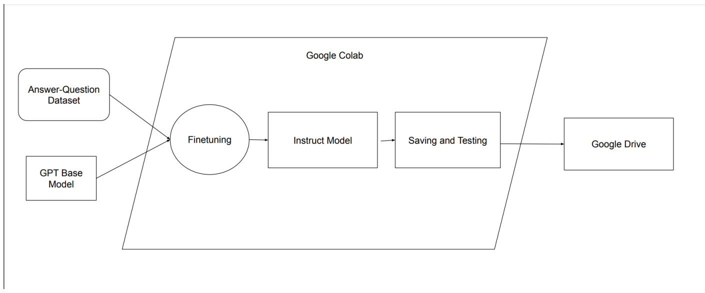

Figure 2 System Design

A. Materials and Dataset

The dataset used in the project was created by combining previously published datasets with AI-generated data. This combined dataset was formatted to fit the structure required by the program. Each row represents a question-answer pair. The data is stored in a text file on Google Drive and retrieved during runtime in Google Colab.

The diversity of the data and the length of the question-answer sentences play a crucial role in effective training. Including quoted words from the question in the answer improves accuracy. Additionally, using longer and more detailed questions and answers during training enhances the model's effectiveness. During testing, longer and richer questions are also preferred to increase the accuracy of the responses generated by the model.

B. Methodology

For the fine-tuning process in the project, methods such as data preparation, BitsAndBytes utilization, and model training were employed. The base GPT model was used as a foundation, with the aim of developing it into an instruct model.

During the data preparation phase, available datasets and AI-generated answers were cleaned and adapted for the project. Incorrect and nonsensical question-answer pairs were removed. The PyTorch framework was utilized for data preparation, using the Datasets library to keep input-output data in tensor format and the DataLoader method to split the data into mini-batches for training [8]. Sequences of varying lengths were equalized using padding tokens, and the data was converted into PyTorch tensors. To meet the project's objective, tokens were reversed before being fed into the model.

BitsAndBytes, a library designed to run deep learning models with low memory consumption, was used to load the model in 8-bit precision [9]. The model was configured using BitsAndBytesConfig, reducing memory usage and enabling it to run with lower VRAM requirements. This optimization ensured efficient use of Google Colab's limited resources.

For flexibility, the PyTorch framework was chosen as the training method. Various parameters were optimized to improve model training. PyTorch, an open-source machine learning library based on the Torch framework, is widely used in natural language processing tasks

IV. Experimental Results

A. Performance Analysis

As shown in Table 1 and Table 2, both models were successfully trained and achieved low train loss values. Model 1 was trained with a learning rate (LR) of 5e-5, starting with an initial train loss of 1.3449 and decreasing to 0.2394. Model 2, on the other hand, was trained with a learning rate (LR) of 1e-4, starting with an initial train loss of 1.4556 and reaching 0.2149. These results indicate that both models effectively learned from the training data.

Table 1 Training Loss and Validation Loss for Model 1

Epoch	Training Loss	Validation Loss
1	1.3449	1.2842
2	1.0447	1.2683
3	0.7768	1.3086
4	0.6974	1.3401
5	0.6472	1.3591
6	0.5772	1.4003
7	0.4606	1.4303
8	0.3912	1.4598
9	0.3112	1.4959
10	0.2394	1.5131

Table 2 Training Loss and Validation Loss for Model 2

Epoch	Training Loss	Validation Loss
1	1.4556	1.4111
2	1.1534	1.3630
3	0.8485	1.4173
4	0.7581	1.4439
5	0.6993	1.4908
6	0.6044	1.5508
7	0.5193	1.5716
8	0.4393	1.6120
9	0.3045	1.6507
10	0.2149	1.7023

As shown in Table 2, the validation loss values of Model 1 were lower and exhibited a slower increase compared to Model 2. The validation loss for Model 1 started at 1.2842 in the first epoch and reached 1.5131 in the final epoch. In contrast, Model 2's validation loss began at 1.4111 and rose to 1.7023 by the final epoch. These results indicate that Model 1 demonstrated better generalization on the validation set and was less affected by overfitting compared to Model 2.

Considering the validation loss values, Model 1, with a lower learning rate, achieved better generalization and produced more successful results on the validation set compared to Model 2. Therefore, it can be concluded that using a smaller learning rate improves generalization performance.

B. Comparative Analysis

Table 3 Evaluation Metrics for Model-1 and Model-2

Metric	**Model-1**	**Model-2**
CIDEr	0.825	0.780
METEOR	0.210	0.198
ROUGE-L	0.342	0.331
BLEU-1	0.388	0.375
BLEU-2	0.252	0.240
BLEU-3	0.168	0.155
BLEU-4	0.102	0.093

BLEU Score: Model-1 achieved higher scores across all BLEU metrics compared to Model-2. This indicates that the sentences predicted by Model-1 showed a greater overlap with the reference sentences, reflecting better precision in its predictions.

ROUGE-L Score: The ROUGE-L metric, which measures the similarity of the generated sentences to reference sentences based on the longest common subsequence, showed that Model-1 outperformed Model-2. This result confirms that Model-1's predictions were more closely aligned with the structure of the reference sentences.

METEOR Score: The METEOR metric evaluates similarity by considering synonymous words and root forms. Model-1's higher METEOR score suggests that its predictions were semantically closer to the reference sentences, demonstrating better understanding and meaning preservation.

CIDEr Score: CIDEr is a critical metric for language generation tasks, emphasizing consensus between generated and reference texts. Model-1 achieved a significantly higher CIDEr score than Model-2, indicating better generalization and alignment with reference texts. The notable difference in CIDEr scores highlights Model-1's superior performance in generating high-quality, contextually relevant sentences.

These results collectively demonstrate that Model-1, with a lower learning rate, achieved better generalization and overall performance compared to Model-2. The use of a smaller learning rate contributed to more accurate and meaningful predictions, making Model-1 the more effective model.

V. Conclusion

In conclusion, in this project, an artificial intelligence model capable of automatically generating questions from given answers was developed. The use of Turkish datasets and fine-tuning methods significantly enhanced the model's performance, ensuring high semantic accuracy and contextual relevance in the generated questions. The outputs obtained during the testing phase demonstrated that the model could consistently produce coherent and meaningful questions based on the provided answers.

The results of this study represent a significant step forward in the field of natural language processing, particularly in the automatic question generation for the Turkish language. Future work could focus on diversifying datasets, training the model on larger and more comprehensive datasets, and expanding the approach to generate questions in multiple languages.

References

[1] C. Yuan and X. Zhang, "Analysis of diversification of intelligent teaching in english literacy integrated classroom empowered by artificial intelligence technology in colleges and universities," *Applied Mathematics and Nonlinear Sciences*, vol. 9, 05 2024.

[2] F. Akyön, D. Çavuşoğlu, C. Cengiz, S. O. Altinuç, and A. Temizel, "Automated question generation and question answering from turkish texts," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 5, p. 1931–1940, 2022. [Online]. Available: https://hdl.handle.net/11511/101785

[3] K. Soleymanzadeh, "Domain specific automatic question generation from text," 01 2017, pp. 82–88.

[4] Y. Kim, H. Lee, J. Shin, and K. Jung, "Improving neural question generation using answer separation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6602–6609, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4629

[5] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, and S. Wang, "Answer-focused and position-aware neural question generation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 3930–3939. [Online]. Available: https://aclanthology.org/D18-1427

[6] K. Kriangchaivech and A. Wangperawong, "Question generation by transformers," *CoRR*, vol. abs/1909.05017, 2019. [Online]. Available: http://arxiv.org/abs/1909.05017

[7] Y.-H. Chan and Y.-C. Fan, "A recurrent BERT-based model for question generation," in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, and D. Chen, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 154–162. [Online]. Available: https://aclanthology.org/D19-5821

[8] S. Shen and C. M. Kittivorawong, "Efficient distributed data loading for large-scale machine learning model training with parax."

[9] Y. Zhao, C.-Y. Lin, K. Zhu, Z. Ye, L. Chen, S. Zheng, L. Ceze, A. Krishnamurthy, T. Chen, and B. Kasikci, "Atom: Low-bit quantization for efficient and accurate llm serving," in *Proceedings of Machine Learning and Systems*, P. Gibbons, G. Pekhimenko, and C. D. Sa, Eds., vol. 6, 2024, pp. 196–209. [Online]. Available: https://proceedings.mlsys.org/paper$_{f}iles/paper/2024/file/5edb57c05c81d04b$
$Paper-Conference.pdf$