KSÜ Mühendislik Bilimleri Dergisi, 28(2), 2025
Araştırma Makalesi

KSU J Eng Sci, 28(2), 2025
Research Article

Kahramanmaras Sutcu Imam University
Journal of Engineering Sciences

Geliş Tarihi : 02.03.2025
Kabul Tarihi :12.04.2025

Received Date : 02.03.2025
Accepted Date : 12.04.2025

EXPLORING THE EFFECTIVENESS OF PRE-TRAINED TRANSFORMER MODELS FOR TURKISH QUESTION ANSWERING

TÜRKÇE SORU CEVAPLAMA İÇİN ÖNCEDEN EĞİTİLMİŞ TRANSFORMER MODELLERİNİN ETKİNLİĞİNİ KEŞFETME

Abdullah Talha KABAKUŞ (ORCID: 0000-0003-2181-4292)

Düzce Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Düzce, Türkiye

Sorumlu Yazar / Corresponding Author: Abdullah Talha KABAKUŞ, talhakabakus@duzce.edu.tr

ABSTRACT

Recent advancements in Natural Language Processing (NLP) and Artificial Intelligence (AI) have been propelled by the emergence of Transformer-based Large Language Models (LLMs), which have demonstrated outstanding performance across various tasks, including Question Answering (QA). However, the adoption and performance of these models in low-resource and morphologically rich languages like Turkish remain underexplored. This study addresses this gap by systematically evaluating several state-of-the-art Transformer-based LLMs on a curated, gold-standard Turkish QA dataset. The models evaluated include *BERTurk*, *XLM-RoBERTa*, *ELECTRA-Turkish*, *DistilBERT*, and *T5-Small*, with a focus on their ability to handle the unique linguistic challenges posed by Turkish. The experimental results indicate that the *BERTurk* model outperforms other models, achieving an F1-score of 0.8144, an Exact Match of 0.6351, and a BLEU score of 0.4035. The study highlights the importance of language-specific pre-training and the need for further research to improve the performance of LLMs in low-resource languages. The findings provide valuable insights for future efforts in enhancing Turkish NLP resources and advancing QA systems in underrepresented linguistic contexts.

Keywords: Artificial intelligence, question answering, transformer, large language model, natural language processing

ÖZET

Doğal Dil İşleme (NLP) ve Yapay Zekâ (AI) alanındaki son gelişmeler, Soru Cevaplama (QA) gibi çeşitli görevlerde olağanüstü performans sergileyen Transformer tabanlı büyük dil modellerinin (LLM'ler) ortaya çıkmasıyla ivme kazanmıştır. Ancak, bu modellerin düşük kaynaklı ve morfolojik açıdan zengin dillerde, özellikle Türkçe'de benimsenmesi ve performansı yeterince araştırılmamıştır. Bu çalışma, özenle hazırlanmış, altın standart bir Türkçe QA veri kümesi üzerinde çeşitli son teknoloji Transformer tabanlı LLM'leri sistematik olarak değerlendirerek bu boşluğu doldurmayı amaçlamaktadır. Değerlendirilen modeller arasında *BERTurk*, *XLM-RoBERTa*, *ELECTRA-Turkish*, *DistilBERT* ve *T5-Small* yer almakta olup, bu modellerin Türkçenin kendine özgü dilsel zorluklarını ele alma yeteneklerine odaklanılmıştır. Deneysel sonuçlar, BERTurk modelinin diğer modellerden üstün performans göstererek 0.8144 F1-skoru, 0.6351 Exact Match ve 0.4035 BLEU skoru elde ettiğini ortaya koymaktadır. Çalışma, dile özgü ön eğitimlerin önemini vurgulamakta ve düşük kaynaklı dillerde LLM performansını artırmaya yönelik daha fazla araştırmaya duyulan ihtiyacı ortaya koymaktadır. Elde edilen bulgular, Türkçe NLP kaynaklarını geliştirme ve yeterince temsil edilmeyen dil bağlamlarında QA sistemlerini ilerletme çabalarına değerli katkılar sunmaktadır.

Anahtar Kelimeler: Yapay zekâ, soru cevaplama, transformer, büyük dil modeli, doğal dil işleme

ToCite: KABAKUŞ, A. T. (2025). EXPLORING THE EFFECTIVENESS OF PRE-TRAINED TRANSFORMER MODELS FOR TURKISH QUESTION ANSWERING. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 28(2), 975-993.

INTRODUCTION

Recent progress in Natural Language Processing (NLP) and Artificial Intelligence (AI) has been driven by the rise of Transformer-based large language models (LLMs), which have demonstrated exceptional performance in diverse tasks, such as Question Answering (QA). QA is the task of predicting a text span within a given paragraph that contains the answer to a specific question. It has become a cornerstone of information retrieval and user interaction, evolving significantly from its early implementations in search engines like *Google*. QA systems enable users to extract precise information quickly, catering to the increasing demand for efficient and accurate query resolution. Moreover, with the rise of Transformer-based models, QA has also emerged as one of the primary methods of interacting with LLMs like *ChatGPT*, *DeepSeek*, *Gemini*, *Grok*, and *Claude*, which leverage advanced understanding of context and semantics to provide detailed and human-like responses. Thanks to these advancements in AI, LLM apps have become one of the hottest topics worldwide, with their introduction being truly groundbreaking. Some recent examples are as follows: *ChatGPT* amassed 100 million users within just 2 months of its launch, becoming the fastest-growing consumer application in history, largely due to its powerful QA capabilities (Hu, 2023). *DeepSeek*, a recently launched LLM app, claimed the No. 1 spot on both the *Google Play Store* and *Apple App Store* in many countries within just two weeks of its initial release (Bonov, 2025; Mehta, 2025). These statistics demonstrate the groundbreaking reception of LLM technologies in the field of Information Technology (IT). These applications are now widely used in various domains, such as customer support, education, healthcare, and virtual assistants, demonstrating their versatility and growing importance in modern technology ecosystems. LLMs have several advantages over humans, including incredible speed, scalability, and consistency in processing vast amounts of data without fatigue. They operate 24/7, provide multilingual capabilities, and have access to extensive knowledge instantly. Unlike humans, they excel in repetitive tasks with minimal errors and maintain objectivity without emotional biases. Their ability to generate, analyze, and summarize information quickly makes them highly efficient for automation, research, and customer support. All of these capabilities stem from unprecedented, groundbreaking advancements in AI technologies, which combine various disciplines, including but not limited to mathematics, statistics, linguistics, electrical & computer engineering, cognitive science, and robotics. The foundations of LLMs can be listed as follows: (i) Neural Networks & Deep Learning, (ii) Transformers, a type of Deep Neural Network (DNN) architecture tailored for sequential data, leveraging self-attention mechanisms to efficiently capture intricate dependencies and relationships in textual data, (iii) tokenization, pre-training & fine-tuning, embeddings to represent words as vectors, (iv) attention mechanism to assign different importance to words, and (v) Reinforcement Learning with Human Feedback to improve responses using human feedback.

Recurrent Neural Networks (RNNs), along with their enhanced versions such as LSTMs (Long Short-Term Memory) and GRUs (Gated Recurrent Units), have proven effective in handling sequential data, demonstrating solid performance in applications like language modeling, speech recognition, and time-series analysis. Unlike RNNs, including LSTMs and GRUs, which process data sequentially and suffer from limitations like vanishing gradients and long training times, Transformers can process entire sequences in parallel, leading to significantly faster training and the ability to capture long-range dependencies more effectively. Transformer-based models, such as *BERT* (*Bidirectional Encoder Representations from Transformers*) (Devlin, Chang, Lee, and Toutanova, 2019), *RoBERTa* (*Robustly Optimized BERT Pre-training Approach*) (Liu et al., 2019), and their multilingual variants, have fundamentally reshaped how machines process and understand human language, even when it is not grammatically perfect. Leveraging self-attention mechanisms, Transformer architectures effectively capture long-range dependencies and contextual relationships, making them particularly adept at complex NLP challenges. While much of the early research focused on widely spoken languages such as English, the adoption and performance of LLMs in low-resource and morphologically rich languages like Turkish remain limited. Turkish, as an agglutinative language, poses unique challenges to NLP systems due to its complex morphology, extensive vocabulary, and word order variability. These linguistic characteristics make it imperative to evaluate and adapt state-of-the-art LLMs for Turkish to ensure their applicability and effectiveness in real-world applications. The QA task is a cornerstone of NLP research and a critical benchmark for evaluating the capabilities of LLMs. It requires models to comprehend a given context and accurately pinpoint relevant answers to specific questions. In recent years, various pre-trained and fine-tuned Transformer models have been introduced for Turkish NLP, such as *BERTurk*, *Multilingual BERT*, and *XLM-RoBERTa* (Conneau et al., 2020). However, a systematic evaluation of their performance on gold-standard Turkish QA datasets is still lacking. Such an evaluation is essential to understand the strengths and limitations of these models and to guide future efforts in improving Turkish NLP resources. This study aims to address this gap by comparing the performance of several state-of-the-art Transformer-based LLMs on a curated, gold-standard Turkish QA dataset. By fine-tuning and evaluating models specifically designed or adapted for Turkish, we seek to provide

comprehensive insights into their capabilities in handling the nuances of the Turkish language. The main contributions of this study are as follows:

- A systematic evaluation of Transformer-based LLMs on a gold-standard QA dataset.
- Highlighting the limited adoption and performance of Transformer-based LLMs in low-resource and morphologically rich languages, with a focus on Turkish.
- A comprehensive assessment of both monolingual and multilingual Transformer-based LLMs for performance comparison in Turkish QA task.
- An in-depth error analysis that uncovers key failure modes in model predictions, including challenges with token span precision, morphological variation, and entity disambiguation in Turkish QA.
- An empirical analysis of computational efficiency, including inference time and GPU memory consumption, to assess the practical applicability of Transformer models in resource-constrained environments.

The rest of the paper is organized as follows: Section 2 provides an overview of related work, Section 3 outlines the materials and methods employed in this study, Section 4 details the experimental results and includes a discussion, and Section 5 concludes the paper with future directions.

RELATED WORK

Early approaches employed rule-based systems, statistical methods, and Information Retrieval (IR) techniques for QA tasks. *Celebi et al.* (Celebi, Gunel, and Sen, 2011) proposed an approach that focuses on processing documents using pattern matching techniques to extract features for a Turkish QA system. The proposed method involves automatically categorizing questions and selecting the correct answer from a predefined answer set. Named Entity Recognition (NER) and pattern-matching are utilized for question categorization, while range queries are employed for submitting questions. A novel approach to ranking similar documents is introduced, replacing traditional distance metrics. *Derici et al.* (Derici et al., 2015) employed a combination of rule-based and statistical approaches to analyze Turkish questions in the geography domain, focusing on focus extraction and question classification. For focus extraction, a rule-based system, termed the Distiller, utilized manually crafted rules over dependency parse trees to identify focus parts, with confidence scores assigned based on expert performance. Additionally, a statistical model, HMM-Glasses, modeled focus extraction as a sequential classification task using a Hidden Markov Model (HMM) and the Viterbi algorithm, leveraging serialized dependency trees in both forward and backward modes to enhance learning diversity. The outputs of these models were combined using weighted confidence scores to determine the final focus parts. For question classification, a rule-based classifier employed unique pattern phrases to assign coarse classes to questions, while a baseline statistical classifier using a TF-IDF-weighted Bag-of-Words (BoW) approach provided a comparison. *Bilgin et al.* (Bilgin, Bozdemir, and Demir, 2024) investigated the performance of LLMs on Turkish QA tasks by fine-tuning five models, namely, (*i*) *bert-base-uncased*, (*ii*) *bert-base-turkish-cased*, (*iii*) *distilbert-base-multilingual-cased*, (*iv*) *mt5-base*, and (*v*) *mBart-large-50*—using a *Turkish SQuAD 1.1* dataset. The research evaluates the models using Exact Match (EM), F1, and Rouge metrics, with *mBART* achieving the highest performance due to its encoder-decoder architecture and multilingual training. The experimental results show that multilingual models generally outperform monolingual ones, while encoder-decoder architectures (e.g., *mBART*, *mT5*) yield better results than encoder-only models like BERT. While *Bilgin et al.*'s work offers valuable insights into model architecture differences and highlights the potential of multilingual transformer models, our study differs in several key aspects. First, we utilize a publicly available gold-standard dataset that includes a distinct structure and richer context-question-answer pairs, enabling a more rigorous evaluation. Second, our study benchmarks a broader range of widely used pre-trained transformer architectures under a unified, reproducible training and evaluation pipeline. Unlike *Bilgin et al.*, who emphasize model-type comparisons, our work additionally incorporates interpretability analysis, inference time comparisons, and model efficiency—crucial aspects for real-world deployment.

In recent years, approaches based on DL and NLP have emerged, providing state-of-the-art solutions for the QA task. *Xu et al.* (Xu, Reddy, Feng, Huang, and Zhao, 2016) proposed a Multi-Channel Convolutional Neural Networks (MCCNNs) model that enhances the learning of robust relation representations by incorporating both lexical and syntactic perspectives. This approach leverages word embeddings as inputs, making it suitable for Knowledge Base Question Answering (KBQA) tasks involving extensive relations within a Knowledge Base (KB). It effectively addresses the data sparsity challenges and improves generalization to unseen words, outperforming traditional feature-based extraction models. *Yu et al.* (Yu et al., 2017) proposed a Hierarchical Residual BiLSTM (HR-BiLSTM)

model designed to enhance relation detection in KB relations by integrating information from two levels: word-level and relation-level representations. To handle unseen relations and utilize lexical semantics, they decomposed relation names into sequences of words, enabling word-level matching between questions and relations. Additionally, they recognized the value of treating relation names as single tokens for better generalization on seen relations, introducing a relation-level matching channel to complement the word-level approach. *Zhu et al.* (Zhu, Cheng, and Su, 2020) introduced a tree-to-sequence method for converting natural language questions into executable queries. They started by constructing candidate queries for a question based on its linked entities. To match these queries with the questions, they utilized an LSTM-based model. For encoding, a tree-based LSTM was employed to capture the contextual structure of entities or relations in a query, effectively encoding the query's structure. A mixed-mode decoder was then used to identify the best query, operating in two modes: the generating mode, which emphasized semantic-level correlations, and the referring mode, which focused on surface-level correlations and language variations. *Luo et al.* (Luo, Lin, Luo, and Zhu, 2018) proposed a neural network-based semantic parsing method that embeds questions and query graphs into a uniform vector space. Query graphs were generated with entity, type, time, and ordinal constraints in five steps, then split into predicate sequences for semantic representation. Global (token-based) and local (dependency path-based) representations were combined using Bi-GRUs, with max pooling applied before cosine similarity calculation. An ensemble approach improved entity linking, and their method encoded entire query graphs instead of word sequences, validated through detailed experiments. *Kotstein* and *Decker* (Kotstein and Decker, 2024) introduced a Transformer encoder model designed for semantic search within Web API documentation, treating the task as a QA problem. Their model matched natural language queries with Web API elements, addressing two tasks: endpoint discovery and parameter matching. Pre-trained BERT models, namely *CodeBERT* and *RoBERTa* were fine-tuned on 1,085,051 samples for parameter matching and 55,659 samples for endpoint discovery, extracted from 2,321 OpenAPI documents. Experimental results showed that CodeBERT slightly outperformed *RoBERTa*, though differences in top-1 accuracy ($\sim$1%) were minimal. Parameter matching performed best with a model fine-tuned on both tasks (81.95% top-1 accuracy), while endpoint discovery achieved its highest accuracy with a task-specific model (88.44% top-1 accuracy). Errors were attributed to missing context in queries, and robustness tests revealed sensitivity to synonyms in domain-specific terms. *Xue et al.* (Xue, Zhang, and Chen, 2024) tackled the challenge of building code compliance checking by introducing a question-answering framework consisting of two main components: (i) a retriever for efficient context extraction from building codes, and (ii) a reader for accurate answer generation. The BM25-based retriever demonstrated strong performance, achieving top-1 precision, recall, and F1-score of 0.95, and top-5 scores of 0.97, 1.00, and 0.99, respectively. The transformer-based reader, utilizing the *xlm-roberta-base-squad2-distilled* model, attained a top-4 accuracy of 0.95 and a top-1 F1-score of 0.84. *Vazrala* and *Khatoon Mohammed* (Vazrala and Khatoon Mohammed, 2025) introduced a Hybrid Gradient Regression-Based Transformer Model (RBTM), which integrates semantic similarity quantification with deep learning methods. Their approach consisted of three main stages: component identification, semantic similarity measurement at both the component and sentence levels, and similarity scoring. The methodology employed tools such as the LemmaChase Lemmatizer for feature extraction, the SNOMED-CT ontology for domain-specific concept identification, and concept2Vec for enhanced vector representations. Additionally, RBTM integrated XGBoost with a transformer architecture to generate similarity scores for answer selection. Evaluated on the *MedQuAD* dataset, the model achieved high performance with 99.09% accuracy, a R^2 score of 97.07%, and an MSE of 0.00227. *Kuligowska* and *Kowalczuk* (Kuligowska and Kowalczuk, 2021) proposed an approach that evaluates various DL models for a QA task, focusing on Recurrent Neural Networks (BiGRU + GloVe + CNN) and fine-tuning *DistilBERT*. The BiGRU model employed early stopping, the *Adam* optimizer, and hyperparameters such as a batch size of 512, 27 epochs, 1-Dimensional (1D) convolution layer with 256 filters and kernel size of 5, two Bidirectional GRU layers with 256 units separated by dropout of 0.1, one-dimensional Global Max Pooling layer, and dense layer with 128 units and Rectified Linear Unit (ReLU) activation function. *DistilBERT* fine-tuning involved batch sizes of 16 or 32, learning rates of $2 \times e^{-5}$ to $5 \times e^{-5}$, and a maximum of 4 epochs, avoiding overfitting and catastrophic forgetting. *DistilBERT* variations included CNN and RNN enhancements. On validation data, the *DistilBERT* ($[CLS]$ token) model achieved the highest Spearman's rank correlation (0.3677), outperforming other models, including BiGRU (0.2817). Using pseudo-labeling, predictions on an unlabeled dataset were incorporated into the training set, further fine-tuning the best-performing *DistilBERT* model for two additional epochs. The pseudo-labeled model significantly improved performance on test data, achieving a Spearman's rank correlation of 0.3866, compared to 0.3785 for the original *DistilBERT*. The findings in light of the experimental results demonstrate that pseudo-labeling and leveraging larger datasets enhance model performance, validating the proposed *DistilBERT*-based approach as superior to other architectures. Unlike previous studies that primarily focus on rule-based or statistical methods for specific domains or rely on traditional DL models, the proposed study systematically evaluates state-of-the-art Transformer-based models on a gold-standard Turkish QA dataset. This

research is critical for advancing the understanding of how modern LLMs perform in low-resource languages, addressing a significant gap in QA systems for underrepresented linguistic contexts. Table 1 provides a comparative overview of key methodologies and their contributions to the QA task, highlighting the evolution from rule-based approaches to advanced transformer-based models.

Table 1. Summary of Pre-Trained Transformer Models Evaluated in This Study. The Table Contextualizes the Comparative Performance Analysis by Presenting Each Model's Architecture, Language Scope, And Tokenization Strategy, Thereby Linking to The Hypothesis That Model Architecture and Language Specialization Significantly Impact QA Performance.

Study	Approach	Key Features	Domain	Performance Highlights
(Celebi et al., 2011)	Rule-based system with NER and pattern matching	Question categorization, predefined answer selection, and range queries	Turkish QA	Introduced novel document ranking to replace traditional distance metrics
(Derici et al., 2015)	Hybrid rule-based and statistical approaches	Focus extraction (rule-based Distiller and HMM-Glasses), question classification with rule-based and statistical methods	Geography	Improved focus identification using a weighted combination of models
(Xu et al., 2016)	Multi-Channel CNN for robust relation representations	Combining lexical and syntactic perspectives, word embeddings for KBQA	Knowledge Base Question Answering	Addressed data sparsity challenges, improving generalization to unseen words
(Yu et al., 2017)	Hierarchical Residual BiLSTM	Word- and relation-level representations, matching unseen relations with lexical semantics	Knowledge Base relations	Enhanced relation detection using hierarchical representations
(Zhu et al., 2020)	Tree-to-sequence model for query conversion	Tree-based LSTM encoder, mixed-mode decoder for query matching	Knowledge Base Question Answering	Effectively encoded query structures, improving correlation-based query selection
(Luo et al., 2018)	Bi-GRU-based semantic parsing for query graph embedding	Combines global and local representations, ensemble approach for entity linking	Knowledge Base Question Answering	Achieved high performance through uniform query graph encoding
(Kotstein and Decker, 2024)	Transformer-based model for semantic search in Web API documentation	Fine-tuned CodeBERT and RoBERTa for parameter matching and endpoint discovery	Web API	CodeBERT outperformed RoBERTa slightly; endpoint discovery achieved 88.44% top-1 accuracy
(Xue et al., 2024)	QA framework with BM25 retriever and transformer-based reader	Combines BM25 for retrieval and xlm-roberta-base-squad2-distilled for reading	Code Compliance	BM25 retriever achieved top-1 precision/recall of 0.95; reader had a top-4 accuracy of 0.95
(Vazrala and Khatoon Mohammed, 2025)	Hybrid Gradient Regression-Based Transformer Model	Semantic similarity scoring using XGBoost and transformers, domain-specific ontology, and embeddings	Medical	Achieved 99.09% accuracy, with low MSE (0.00227)
(Kuligowska and Kowalczuk, 2021)	Evaluation of BiGRU + GloVe + CNN and DistilBERT-based models for QA	Fine-tuned DistilBERT with pseudo-labeling and dataset augmentation	General QA	DistilBERT achieved the best Spearman's rank correlation (0.3866 with pseudo-labeling)

MATERIAL AND METHOD

Python programming language was used for all programming aspects of this study. More specifically, the *transformers* (Wolf, Debut, Sanh, Chaumond, and ..., 2020), a widely used Python package developed by *Hugging Face*, was employed to train state-of-the-art LLMs, particularly transformer architectures, such as *BERT*, *GPT (Generative Pre-trained Transformer)*, *RoBERTa*, *Llama* (Large Language Model Meta AI), and *T5* (*Text-to-Text Transfer Transformer*). In addition to the *transformers* library, the *evaluate* ("Evaluate," 2025) package developed by *Hugging Face* was also employed to assess model performance in a more straightforward and standardized manner. *PyTorch* (Paszke et al., 2019) was utilized as the DL backend because of its dynamic computation graph, offering flexibility in model design and facilitating rapid experimentation. This adaptability is essential for developing and fine-tuning complex architectures, such as Transformers, as applied in the proposed study. Additionally, *PyTorch*'s seamless GPU support via CUDA ensures efficient handling of large-scale computations. *Scikit-learn* (Pedregosa et al., 2011) and *Pandas* (The pandas development team, 2020), two widely used Python packages were employed for data analysis and data manipulation. *Matplotlib* (Hunter, 2007) and *seaborn* (Waskom,

2021) were employed for visualization purposes, including generating model training plots and conducting exploratory data analysis. All the experiments conducted in this study were performed on *Kaggle*, as it offers a robust, powerful, and efficient environment for developing ML models. In the following subsections, the utilized dataset, the employed Transformer-based LLMs, and evaluation metrics are described, respectively. Fig. 1 illustrates an overview of the workflow of the proposed approach. The process begins with loading the dataset, followed by tokenizing the text to prepare it for model input. Next, an appropriate Transformer model is selected and fine-tuned using the training data. Finally, the fine-tuned model is evaluated using key performance metrics to assess its effectiveness in answering questions accurately.

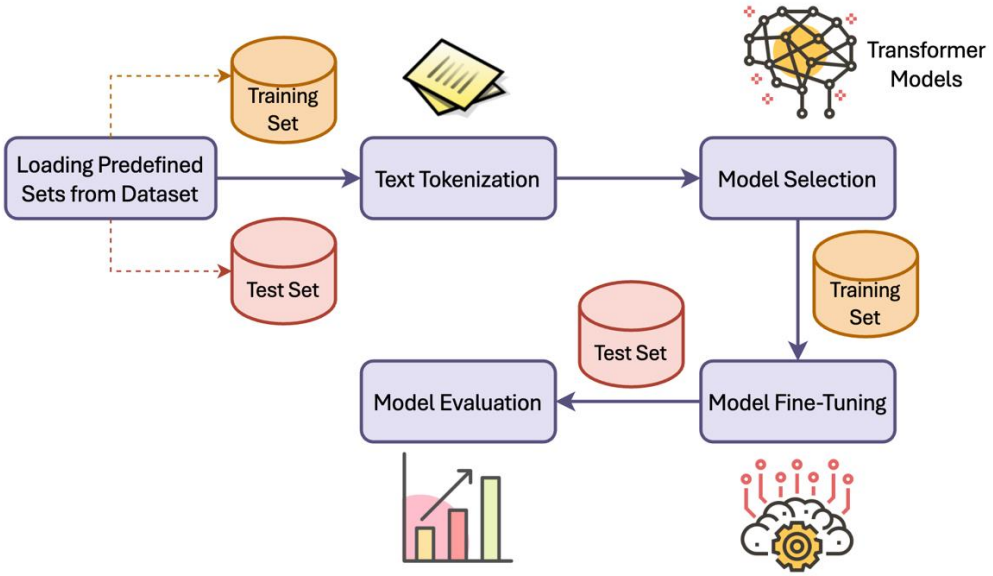

Figure 1. Overview Of the Turkish QA Pipeline Used in This Study. This Architecture Illustrates the Flow from Input (Context and Question) To Answer Prediction Using a Pre-Trained Transformer Model. It Highlights the Central Hypothesis That Fine-Tuning Existing Multilingual or Monolingual LLMs On a Structured, Gold-Standard Turkish Dataset Can Yield High Accuracy in Turkish QA Tasks.

Dataset Description and Data Preprocessing

Using a gold-standard dataset enhances the reliability, comparability, and credibility of the study's results, allowing for accurate benchmarking and validation against established standards. Therefore, we utilized a publicly available gold-standard Turkish QA dataset (Soygazi, Ciftci, Kok, and Cengiz, 2021) consisting of two JSON files: one for training and one for evaluation. The training and evaluation sets consisted of 14,221 and 3,114 question-answer pairs, respectively. As these sets are predefined by the dataset providers, no additional splitting or shuffling was performed, thereby maintaining consistency with the benchmark configuration. The structure of the data within these JSON files is as follows: Each subject (referred to as "*title*") contains multiple contexts. Under each context, there is one or more questions paired with their respective answers. To align with the common practice in ML, we converted these JSON files into CSV format. The resulting CSV files include the following fields: (*i*) "*question*" stores each individual question, (*ii*) "*answer*" contains the respective answer for each question, and (*iii*) "*context*" holds the context relevant to the question. The average lengths of the "*question*," "*answer*," and "*context*" fields are 66, 26, and 1,239 characters, respectively. For a clearer understanding of the variability and typical sizes of the dataset entries, the length distributions of these fields are visualized in Fig. 2.

The dataset comprised 652 unique subjects from Turkish & Islamic Science History. Some sample subjects are given (*in English*) as follows: "*Science in Anatolia*", "*Clocks*", "*Ibn Khaldun*", "*Platon*", "*Treaty of Paris*", and "*Science in Central Asia*". Some sample questions, answers, and their corresponding contexts from this dataset are provided in Table 2.

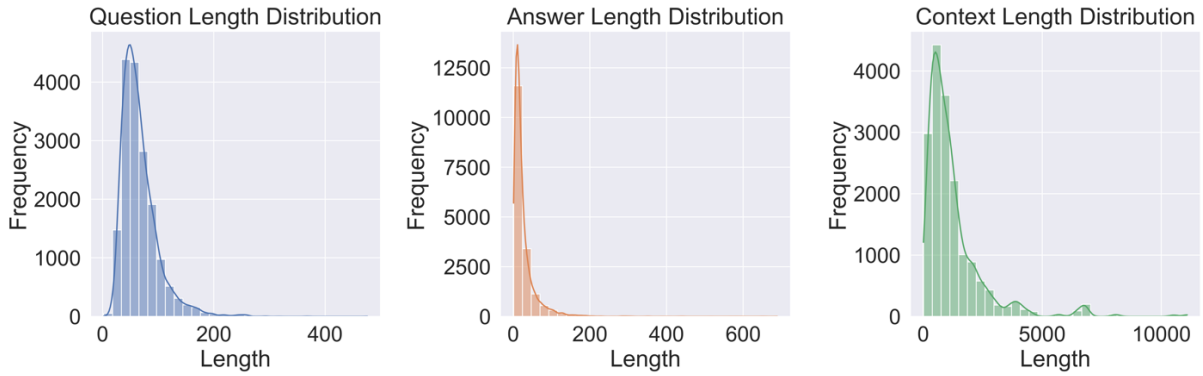

Figure 2. The Length Distributions of the Fields Available in the Dataset, Illustrating the Character Counts for the "*Question*," "*Answer*," and "*Context*" Fields, from Left to Right. These Plots Reveal the Variability in Input Sizes, Supporting the Challenge That Transformer-Based Models Must Generalize Across Diverse Linguistic Structures and Lengths, Especially in Morphologically Rich Languages Like Turkish.

As described, each sample in the dataset consists of a triplet: (***i***) *context*, (***ii***) *question*, and (***iii***) *answer*. We utilized the *Hugging Face transformer*'s *AutoTokenizer* class to tokenize the *question* and *context* together, with truncation enabled and maximum length set to **512** tokens. Answers were mapped back to token positions by first identifying their character start index in the context and then using offset mappings to locate the corresponding start and end token positions. If an answer span could not be precisely aligned due to truncation or tokenization edge cases, the sample was excluded to maintain data quality.

Employed Models

For the task of Turkish QA, we utilized several transformer-based models pre-trained on large-scale datasets. These models have shown strong performance in various NLP tasks, including QA, and are well-suited for processing Turkish text due to their language-specific adaptations and multilingual capabilities. Some of these pre-trained models come in two versions based on case-sensitivity: (i) cased, and (ii) uncased. Since the dataset we used is case-sensitive, we intentionally chose the cased versions of the pre-trained models. One more criterion for model selection was the use of both monolingual and multilingual Transformer-based LLMs to compare their performance in the Turkish QA task. All models were evaluated on the identical, predefined training and testing sets from the gold-standard Turkish QA dataset as using fixed splits ensures direct comparability across models, eliminating variability from random partitioning. This approach aligns with standard practices in QA benchmarking (e.g., *SQuAD* evaluations). We employed the *AutoModelForQuestionAnswering* class from the *Hugging Face*'s *transformers* library, which wraps each model with a span-based question answering head — typically composed of two linear layers projecting the transformer output to start and end logits. Five transformer models were evaluated: (i) *ELECTRA-Turkish*, (ii) *XLM-RoBERTa*, (iii) *BERTurk*, (iv) *DistilBERT*, and (v) *T5-Small*. While the first four are encoder-only models with span prediction heads, *T5-Small* uses an encoder-decoder architecture to generate free-form answers in a sequence-to-sequence manner. The employed models within the scope of this study are described in the following subsections.

ELECTRA-Turkish

ELECTRA (*Efficiently Learning an Encoder that Classifies Token Replacements Accurately*) (Clark, Luong, Le, and Manning, 2020) is a state-of-the-art transformer model that improves upon *BERT* by using a more sample-efficient pre-training approach. The *electra-base-turkish-cased-discriminator* (MDZ Digital Library team, 2024b) (hereafter referred to as *ELECTRA-Turkish*) model is a version of *ELECTRA* pre-trained proposed for Turkish. Unlike traditional masked language models like *BERT*, *ELECTRA* generates corrupted tokens and trains the model to distinguish between real and fake tokens. This makes it more efficient and capable of learning better contextual representations, which is crucial for tasks like QA in Turkish, where understanding subtle linguistic nuances is essential.

Table 2. Some Sample Questions, Answers, and Their Contexts from the Utilized Dataset (Translated to English and the Original Turkish Text are Given in Parentheses).

Question	Answer	Context
When was the Battle of Otlukbeli fought? (TR: *Otlukbeli Savaşı hangi tarihte yapıldı?*)	On August 11, 1473 (TR: *11 Ağustos 1473'te*)	In the face of the growing power of the Ottoman Empire, the Karamanids allied with the Aq Qoyunlu in Eastern Anatolia. In 1466, Mehmed the Conqueror launched a new Anatolian campaign and captured Konya, the capital of the Karamanids. However, after his return to Istanbul, the Karamanids regained the territories that had fallen to the Ottomans. Gedik Ahmed Pasha, who would later become grand vizier, defeated the Karamanids once again in 1471. The Aq Qoyunlu continued to support the Karamanids. On August 11, 1473, the Ottoman forces inflicted a heavy defeat on the Aq Qoyunlu ruler Uzun Hasan at the Battle of Otlukbeli. In 1474, the Karamanid Principality was completely eliminated. (TR: *Osmanlı Devleti'nin gelişen bu gücü karşısında Karamanoğulları, Doğu Anadolu'daki Akkoyunlular'la ittifak kurdu. Fatih, 1466'da yeni bir Anadolu seferine çıktı.Karamanoğullarının başkenti Konya'yı ele geçirdi.Ama İstanbul'a dönünce Karamanoğulları, Osmanlılara geçen yerleri geri aldılar. Sonradan sadrazam olacak olan Gedik Ahmed Paşa 1471'de Karamanoğullarını bir kez daha yenilgiye uğrattı.Akkoyunlular, Karamanoğullarını desteklemeye devam ettiler. 11 Ağustos 1473'te Otlukbeli Savaşı'nda Akkoyunlu hükümdarı Uzun Hasan'ı ağır bir yenilgiye uğrattı. 1474 yılında Karamanoğulları Beyliği'ni tamamen ortadan kaldırdı.*)
Who drew the first world map? (TR: *İlk dünya haritasını kim çizdi?*)	Piri Reis	The Italian navigator Christopher Columbus, who discovered the American continent, presented his transoceanic voyage, which he had been planning for about 14 years, to the King of Portugal in 1484, but it was rejected. Unable to find a financial backer, Columbus faced financial difficulties and engaged in trade between Europe and the Ottoman Empire. During this period, in 1484, he applied to Sultan Bayezid II with a priest. The Sultan did not take this eccentric man seriously and rejected his request. Two years after Bayezid's refusal, Columbus approached the King and Queen of Spain and unknowingly discovered America in 1492. Mistaking the land he reached for India, he called the native people 'Indians.' Years later, a Spaniard who had accompanied Columbus on three voyages to America was captured by Piri Reis's uncle, Kemal Reis, after a battle. The Spaniard gave Kemal Reis a map of the American coasts discovered by Columbus. Based on the information from this map, Piri Reis drew the first world map in 1513. (TR: *Amerika kıtasını keşfeden İtalyan denizci Kristof Kolomb, yaklaşık 14 yıldır tasarladığı okyanus ötesinde yolculuğu 1484'te Portekiz Kralına sundu ama reddedildi. Destekleyecek bir finansör bulamayınca maddi zorluklara giren Kolomb, Avrupa ile Osmanlı arasında ticaret ile uğraştı. Bu dönemde, 1484'te Sultan II. Bayezid'e bir papaz eşliğinde başvurdu.Sultan, karşısına çıkan bu delidolu insanı ciddiye almadı ve talebini reddetti.Kolomb, Bayezid'den 2 yıl sonra İspanyol kral ve kraliçesine müracaat etti, ve 1492'de de Amerika'yı farkında olmadan keşfetti. Geldiği yeri Hindistan zannederek karşılaştığı halka Hindistanlılar 'Indian' dedi.İlerki yıllarda Kolomb ile 3 kez Amerika'ya gitmiş bir İspanyol, bir savaş sonrasında Piri Reis' in amcası Kemal Reis'e esir düştü ve Kolomb'un keşfettiği Amerika kıyılarının haritasını amcasına verdi. Piri Reis bu haritadaki bilgilerden yola çıkarak 1513'de ilk dünya haritasını çizdi.*)
In which year was the Italian translation of the work by Al-Idrisi made? (TR: *El-İdrisi'ye ait olan eserin İtalyanca çevirisi hangi yılda yapılmıştır?*)	1600	Another astonishing fact—unlike the maps—is that the text of Al-Idrisi's work, mentioned above, became known in Europe in a late, highly abridged, and even almost distorted edition. This text was first published in 1592 in Rome, then translated into Italian by B. Baldi in 1600, and into Latin by two Maronites, Gabriel Sionita and Johannes Hesronita, in 1619. However, the Latin translation was published under the title *Geographie Nubiensis* (Geography of the Nubian), without mentioning the author Al-Idrisi, and for a long time, it was cited in this way. While Arab-Islamic human geography remained largely unknown outside of Spain in Europe for a long time, today we can undoubtedly trace how mathematical geography and cartography from the Arab-Islamic cultural sphere profoundly influenced European successors from the 11th to the 18th century. (TR: *Yine hayrete düşüren bir diğer husus –haritaların aksine– yukarıda bahsedilen el-İdrisi'ye ait eser metninin geç dönemde ve aşırı kısaltılmış, hatta neredeyse tahrif edilmiş bir redaksiyonla Avrupa'da tanınmış olmasıdır. Bu metin ilkin 1592 yılında Roma'da basıldı ve 1600 yılında B. Baldi tarafından İtalyanca'ya ve 1619 yılında iki Maronit Gabriel Sionita ve Johannes Hesronita tarafından Latince'ye çevirildi. Fakat Latince çeviri, yazar elİdrisi adı anılmaksızın, Geographie Nubiensis (Sudanlının Coğrafyası) diye yayınlandı ve uzunca bir süre bu şekilde alıntılandı. Arap-İslam beşeri coğrafyası geniş ölçüde ve uzun zaman İspanya dışı Avrupa'da bilinmemiş olarak kaldıysa da, bugün biz kuşkusuz, Arap-İslam kültür çevresine ait matematiksel coğrafya ve kartografyanın 11. yüzyıldan 18. yüzyıla kadar Avrupalı ardıllarını çok derinden etkilediğini tespit edebiliyoruz.*)

XML-RoBERTa

XLM-RoBERTa is a multilingual transformer model pre-trained in a vast number of languages, including Turkish. It is based on *RoBERTa* (Liu et al., 2019), an optimized version of BERT that removes the Next Sentence Prediction (NSP) objective. *XLM-RoBERTa* is designed to work across multiple languages, and its multilingual nature would allow it to generalize well for Turkish QA tasks, even if the model was primarily trained on other languages, making it a robust choice for multilingual or cross-lingual QA systems.

BERTurk

The *bert-base-turkish-cased* (*a.k.a. BERTurk*) (MDZ Digital Library team, 2024a) model is a pre-trained version of BERT proposed for Turkish. BERT has set new benchmarks in NLP tasks like text classification, Named Entity Recognition (NER), and QA. The BERT architecture enables a strong contextual understanding of Turkish text, making it highly suitable for extracting answers from Turkish text-based documents.

DistilBERT

DistilBERT (Sanh, Debut, Chaumond, and Wolf, 2019) is a lighter, faster version of *BERT*, offering a smaller model size with comparable performance. It is distilled from the *BERT* architecture, reducing the number of parameters while retaining much of *BERT*'s capabilities. *DistilBERT* is particularly useful for QA tasks where computational efficiency and inference speed are critical. Given its smaller size, it is a practical choice for deploying real-time question-answering systems in Turkish, while still leveraging the core strengths of *BERT* for understanding and extracting answers from the text.

T5-Small

T5-Small is a lightweight variant of the *T5* model (Raffel et al., 2020), designed for various NLP tasks using a unified text-to-text approach. Unlike encoder-only models like BERT or decoder-only models like GPT, *T5* employs an encoder-decoder architecture, making it particularly effective for generative tasks such as text summarization, translation, and QA. With approximately 60M parameters, *T5-Small* is computationally efficient while maintaining strong performance across multiple languages, including Turkish. A comparison of the employed models is given in Table 3.

Table 3. Comparative Overview of The Transformer-Based Models Employed for The Turkish QA Task. This Table Highlights the Architectural Differences, Language Specialization, And Parameter Sizes of Each Model, Along with Their Strengths and Weaknesses in Handling Turkish QA Challenges. It Supports the Study's Hypothesis That Monolingual Models with Language-Specific Pre-Training (e.g., *BERTurk*, *ELECTRA-Turkish*) May Offer Better Linguistic Understanding for Turkish, While Lighter or Multilingual Models Provide Advantages in Efficiency and Cross-Lingual Generalization.

Model	Architecture	Language	Size	Strengths	Weakness
ELECTRA-Turkish	ELECTRA	Turkish	$\sim 135M$ parameters	Efficient pre-training, better token-level understanding, and high contextual representation.	Requires more resources for fine-tuning compared to lighter models.
XLM-RoBERTa	RoBERTa	Multilingual	$\sim 270M$ parameters	Strong cross-lingual performance, robust for multilingual and limited-resource tasks.	Larger size leads to higher computational demands, and less optimized for Turkish-specific nuances.
BERTurk	BERT	Turkish	$\sim 110M$ parameters	Captures Turkish grammar and semantics well, proven performance in various NLP tasks.	May underperform in tasks requiring case sensitivity; relatively resource-intensive.
DistilBERT	Distilled BERT	Multilingual	$\sim 66M$ parameters	Faster inference, lower resource requirements, suitable for real-time applications.	Lower accuracy compared to full-sized BERT models, lacks Turkish-specific pre-training.
T5-Small	T5	Multilingual	$\sim 60M$ parameters	Strong generative capabilities, effective for text-to-text tasks, lightweight compared to larger T5 variants.	Lower performance on specialized tasks; may require fine-tuning for Turkish-specific applications.

Evaluation Metrics

We employed a range of evaluation metrics to comprehensively assess the performance of the transformer-based language models on the Turkish QA task. These metrics provide quantitative insights into the effectiveness of the models in understanding and responding to questions in Turkish. The metrics used in this study are described in the following subsections.

Exact Match

The Exact Match (EM) metric calculates the proportion of questions where the predicted answer precisely matches the ground truth answer. It is a stringent metric that ensures the model provides accurate answers without any deviation. The equation for EM is given in Eq. 1, where N is the total number of questions, and $\mathbb{1}$ is the indicator function that equals 1 when the prediction is identical to the ground truth and 0 otherwise.

$$EM = \frac{1}{N}\sum_{1}^{N} \mathbb{1}(Prediction_i = Ground\ Truth_i) \tag{1}$$

F1-Score

The F1-Score assesses the similarity between the predicted and ground truth answers by balancing precision and recall. It is particularly useful when partial matches between answers are significant, as it accounts for token-level correctness. To define these metrics more clearly, we first introduce the following terms: T (True Positives) refers to the tokens that are correctly predicted as part of the answer; N (True Negatives) refers to the tokens that are correctly predicted as not part of the answer; P (predicted Positives) represents the tokens predicted by the model as part of the answer, regardless of whether they are correct; and F (False Positives) represents the tokens that are incorrectly predicted as part of the answer. Precision is defined as the fraction of predicted positive tokens that are also correct. The equation for Precision is provided in Eq. 2. Recall, on the other hand, measures the fraction of true positive tokens that are correctly predicted, and its equation is given in Eq. 3. Precision quantifies the fraction of tokens in the predicted answer that also appear in the ground truth, while Recall measures the fraction of tokens in the ground truth that are present in the prediction. The F1-Score is the harmonic mean of Precision and Recall, with its equation given in Eq. 4.

$$Precision = \frac{T}{P + F} \tag{2}$$

$$Recall = \frac{T}{T + N} \tag{3}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

In our evaluation, we used the *load_metric("f1")* function from the *Hugging Face*'s *Datasets* ("Datasets," 2025) library, which computes the F1-score based on the token-level overlap between the predicted answer and the ground truth. Specifically, this implementation calculates Precision and Recall by comparing the number of overlapping tokens in the predicted and reference answers. The F1-score is then computed as the harmonic mean of these two values. It is worth mentioning that this metric does not rely on exact string matching and allows for partial matches when the predicted answer contains relevant words from the ground truth. While this method does not perform semantic similarity assessment like BLEU or embedding-based metrics, it offers a balance between exactness and flexibility, especially in extractive or short-form QA tasks where small variations can occur due to tokenization or model generation behavior.

BLEU Score

BLEU (Bilingual Evaluation Understudy) (Papineni, Roukos, Ward, and Zhu, 2002) measures the similarity between the predicted and ground truth answers by comparing n-grams. It is commonly used in NLP tasks to evaluate the quality of generated text. The equation for the BLEU score is given in Eq. 5, where BP is the brevity penalty, w_n are weights for n-grams, and p_n is the precision for n-grams of size n.

$$BLEU = BP \times exp\left(\sum_{n=1}^{N} w_n log p_n\right) \quad (5)$$

EXPERIMENTAL RESULTS AND DISCUSSION

The models employed in the study were trained on the training set using the same configuration to ensure a fair environment for benchmarking their QA performance. All experiments were conducted using the Hugging Face transformers and datasets libraries in a Kaggle notebook environment with GPU ($T4 \times 2$) acceleration. For each model, we employed a consistent training pipeline to ensure fair comparison. The models were fine-tuned using the *AdamW* (Loshchilov and Hutter, 2019) optimizer with a learning rate of $5 \times e^{-5}$, a batch size of 4 for both training and evaluation, and a total of 3 epochs. To prevent overfitting, early stopping was applied with patience of 2 evaluation steps based on the F1-score on the validation set. Gradient clipping with a maximum norm of 1.0 was used to improve training stability. Additionally, a linear learning rate scheduler with warm-up steps equal to 10% of the total training steps was adopted. To monitor model performance, evaluation was performed at the end of each epoch, and logging occurred every 10 steps to track loss and training progress. All models were trained using mixed-precision (*FP16*) to accelerate computation and reduce memory usage. The *predict_with_generate* parameter was enabled to facilitate text generation-based tasks, such as QA. After the models were fine-tuned on the QA training set, they were evaluated on the test set. According to the experimental results, the best-performing model in terms of QA capability was *BERTurk*, achieving an F1-score of 0.8144, an EM of 0.6351, and a BLEU score of 0.4035. Meanwhile, *DistilBERT* was found to be the most lightweight model in terms of inference time, with an inference time of 31 milliseconds, while *T5-Small* was the most lightweight in terms of training duration, with a training duration of 3,005 seconds. However, *T5-Small* exhibited much more limited QA capabilities compared to *BERTurk*. The evaluation results of the employed models in terms of the utilized evaluation metrics are given in Table 4. The *BERTurk* model outperformed *DistilBERT*, *XLM-RoBERTa*, *ELECTRA-Turkish*, and *T5-Small* due to its Turkish-specific pretraining, full-sized architecture, and robust Masked Language Model (MLM) approach. Unlike *XLM-RoBERTa* and *DistilBERT*, which are trained on multilingual corpora, *BERTurk* is pre-trained exclusively on Turkish text, allowing it to better capture the language's morphological richness and agglutinative structure. Additionally, compared to *DistilBERT*'s lighter, distilled architecture, the full-sized *BERT* model retains deeper contextual representations, which are crucial for complex QA tasks. While *ELECTRA-Turkish* employs a replacement-based pretraining strategy, its discriminator approach may not generalize as effectively as BERT's MLM for extractive QA tasks, where understanding masked words enhances answer extraction, even though the employed model was specifically proposed for Turkish. Moreover, *T5-Small* follows a Sequence-to-Sequence (Seq2Seq) paradigm, which is more suited for generative tasks rather than extractive QA, further contributing to its lower performance. Finally, *BERTurk* benefits from a tokenizer specifically optimized for Turkish grammar, case sensitivity, and subword structures, ensuring better token alignment and improving answer extraction accuracy. The combination of these factors makes *BERTurk* the most effective model for Turkish QA, outperforming its counterparts. The performance comparison of the employed models, based on F1-Score, EM, and BLEU Score metrics, is illustrated in Fig. 3 to visually highlight the strengths and weaknesses of each model.

We analyzed tokenization and prediction behaviors for agglutinative Turkish words to illustrate how morphological complexity impacts model performance. Turkish relies heavily on suffix stacking (e.g., a single word like "*okullarımızdaki*" ("*in our schools*" in English) combines $root + plural + possession + location + relational\ suffixes$). While *BERTurk*'s Turkish-specific tokenizer preserves semantic coherence by segmenting morphemes contextually, multilingual models like *XLM-RoBERTa* often fragment suffixes into suboptimal units, degrading answer precision. Table 5 demonstrates this contrast using examples from our dataset, correlating with *BERTurk*'s superior performance over *XLM-RoBERTa* across all the evaluation metrics used.

To validate the significance of the observed performance differences between models, we conducted a one-way ANalysis Of Variance (ANOVA) on the F1-scores obtained across five independent runs for each model. The ANOVA test yielded a p-value of $p < 0.01$, indicating that the differences in F1-score across models are statistically significant. Post-hoc Tukey's Honestly Significant Difference (HSD) tests further confirmed that *BERTurk* significantly outperforms all other models ($p < 0.05$), especially compared to lightweight models such as *DistilBERT* and *T5-Small*. Variance analysis also showed that *BERTurk*'s performance is consistent across runs, with a standard deviation of only ± 0.0042 in the F1-score, compared to ± 0.0176 for *XLM-RoBERTa*. These statistical evaluations reinforce the reliability and superiority of *BERTurk* in the context of Turkish QA.

Table 4. Evaluation Results of The Transformer-Based Models Used in The Turkish QA Task, Highlighting Their Training and Inference Efficiency Alongside Core Performance Metrics. All Metrics were Averaged Over Five Independent Runs. F1-Score and EM Assess Answer Overlap and Precision, While BLEU Score Reflects Fluency and Syntactic Similarity. This Comparison Reveals That *BERTurk* Provides the Best Balance of Performance and Moderate Computational Cost, Supporting the Study's Hypothesis That Monolingual Models Fine-Tuned for Turkish Outperform Multilingual or Generative Counterparts in This Domain.

Model	Training Duration (s)	Inference Time (ms)	F1-Score	EM Score	BLEU Score
DistilBERT	3,139	**31**	0.6615	0.4494	0.3051
BERTurk	5,452	46	**0.8144**	**0.6351**	**0.4035**
XLM-RoBERTa	6,563	53	0.7804	0.6020	0.3665
ELECTRA-Turkish	5,196	43	0.8009	0.6281	0.3974
T5-Small	**3,005**	53	0.1207	0.0437	0.0385

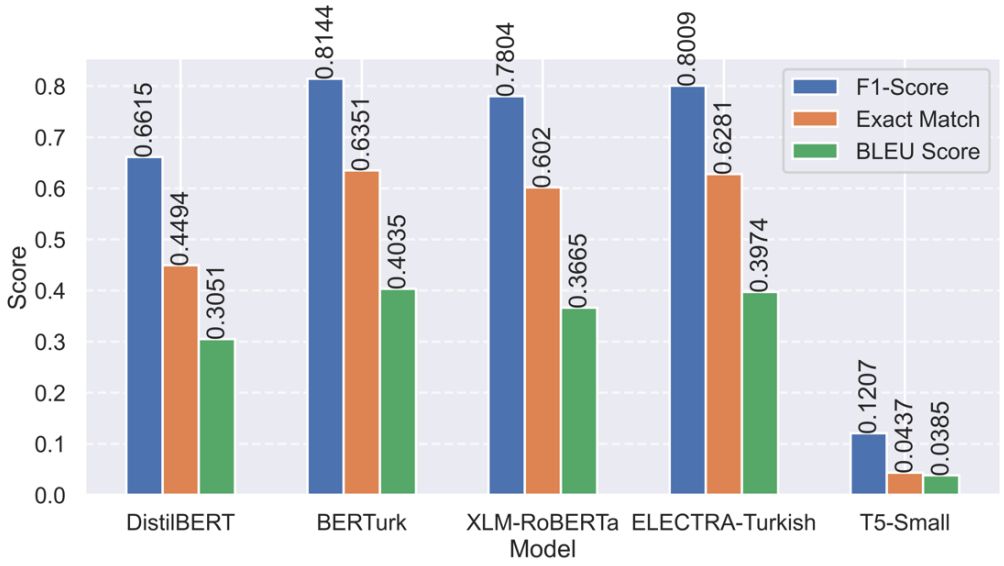

Figure 3. Comparison of the Employed Models Based on the Evaluation Metrics, Namely F1-Score, EM, and BLEU Score. *BERTurk* Demonstrates Superior Performance Across All Metrics, Significantly Outperforming *DistilBERT*, *XLM-RoBERTa*, *ELECTRA-Turkish*, And *T5-Small*. Example Case Showing the Model's Performance on A Long and Contextually Complex Turkish Question. This Visual Supports the Study's Claim That Well-Fine-Tuned LLMs Can Capture Semantic Dependencies in Longer Contexts, Despite the Syntactic Richness of Turkish.

Table 5. Morphological Analysis of Model Performance on Agglutinative Turkish Structures. This Table Compares How *BERTurk* and *XLM-RoBERTa* Tokenize and Process Morphologically Complex Turkish Words, Highlighting the Challenges Posed by Agglutination (e.g., Suffix Stacking). The Examples Demonstrate That *BERTurk*'s Turkish-Specific Tokenization Preserves Semantic Coherence by Segmenting Morphemes Contextually, Whereas *XLM-RoBERTa*'s Multilingual Tokenizer Often Fragments Suffixes into Non-Meaningful Units.

Example	Morphological Breakdown	BERTurk Output	XLM-RoBERTa Output
"Okullarımızdaki kitaplar" (EN: "*Books in our schools*")	*"okul" + "lar" + "ımız" + "da" + "ki"* (EN: "*school*" + *PL* + "*our*" + *LOC* + *REL*)	Correct (context-aware)	Incorrect (fragmented)

The evaluation of training duration and inference time provides crucial insights into the computational efficiency and real-world applicability of the models. *T5-Small* demonstrated the fastest training time at 3,005 seconds, showcasing its lightweight architecture optimized for speed. Conversely, *XLM-RoBERTa* exhibited the longest training duration at 6,563 seconds, likely due to its multilingual corpus and more complex architecture. In terms of inference time, *DistilBERT* outperformed all models with the fastest inference time of 40 milliseconds, aligning with its design for resource-efficient deployments. However, despite its efficiency, *DistilBERT*'s predictive performance was considerably lower compared to *BERT-Turkish*, which achieved the highest evaluation scores but had a slightly longer inference time of 46 milliseconds. This trade-off between speed and accuracy is evident across models. While *ELECTRA-Turkish* offered relatively faster inference (43 milliseconds) and moderate training time, its performance did not match the top-performing models. *T5-Small*, despite its swift training phase, showed slower inference (53 milliseconds) and lower evaluation metrics, suggesting that it is more suitable for generative tasks rather than extractive question-answering. The obtained training and inference times of the employed models are illustrated in Fig. 4, highlighting the trade-offs between computational efficiency and predictive performance across models.

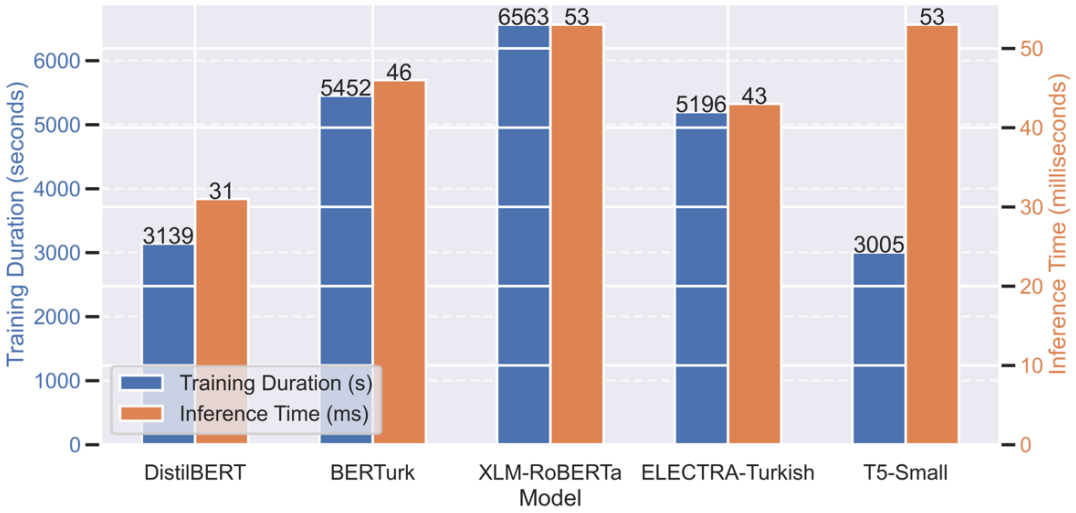

Figure 4. Comparative Analysis of Training Duration and Inference Time Across the Evaluated Transformer-Based Models. The Bar Chart Highlights Efficiency Trade-Offs by Displaying Each Model's Resource Requirements in Terms of Training and Inference, Which Is Critical for Determining Real-World Applicability in Low-Resource or Latency-Sensitive Turkish QA Deployments.

In addition to reporting the training duration and inference time for each model, we also evaluated their practical deployment aspects by assessing memory consumption. Using Python's *tracemalloc* and *psutil* libraries, we measured peak memory usage during inference. The results show that *BERTurk* and *XLM-RoBERTa*, while achieving strong performance in terms of F1 and EM scores, exhibited higher memory consumption (approximately 1 GB of GPU memory). This could pose challenges for deployment in memory-constrained environments, such as on mobile or edge devices. On the other hand, *T5-Small* and *ELECTRA-Turkish* demonstrated significantly lower GPU memory footprints (243 MB and 433 MB, respectively), making them more suitable for lightweight applications, despite their relatively lower performance. These findings underline the trade-off between model accuracy and computational efficiency, providing valuable insights for selecting models based on specific application needs. The bar plot illustrating the GPU memory consumption (in MB) of the employed transformer models during inference is presented in Fig. 5.

Table 6 presents a comparison between the ground truth answers and the answers generated by the fine-tuned *BERTurk* model for the given sample contexts and questions from the test set. Each row in the table contains a context, a corresponding question, the ground truth answer, and the answer generated by the fine-tuned *BERTurk* model. This comparison demonstrates the model's ability to accurately retrieve and generate answers based on the provided contextual information, showcasing its performance in understanding and answering factual questions.

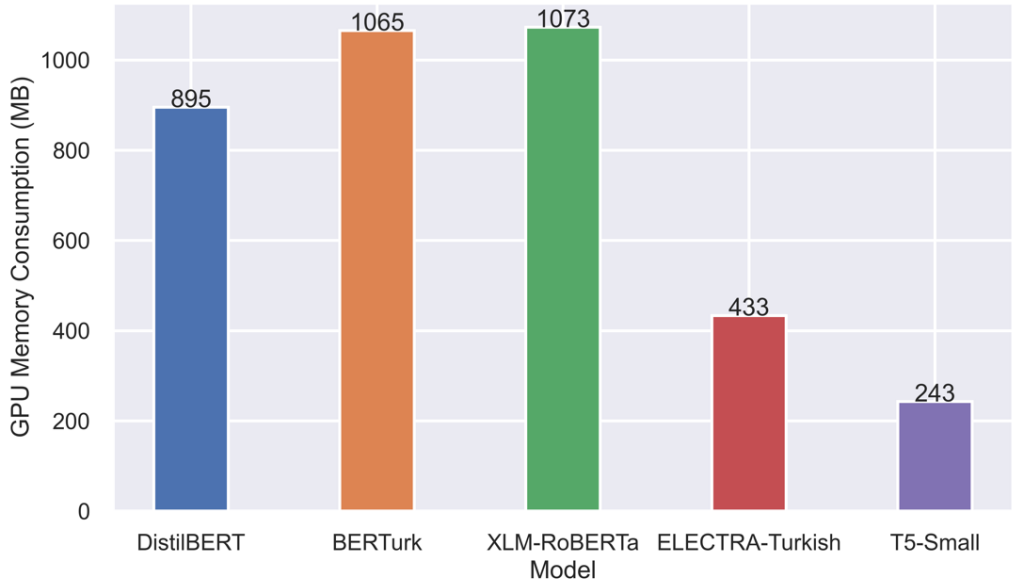

Figure 5. GPU Memory Consumption (In MB) of the Evaluated Transformer Models. The Results Highlight the Variations in Memory Consumption Across the Models, With *BERTurk* and *XLM-RoBERTa* Consuming Significantly More Memory Compared to The Lighter Models, Such as *DistilBERT* and *T5-Small*, Which Are More Suitable for Deployment in Memory-Constrained Environments Like Mobile or Edge Devices.

Table 6. Comparison of Ground Truth and Answer Generated by the Fine-Tuned *BERTurk* for the Given Sample Contexts and Questions from the Test Set (Translated to English and the Original Turkish Text are Given in Parentheses).

Context	**Question**	**Ground Truth**	**Answer Generated by BERTurk**
Ibn Kathir (1301 - 1373) was a Syrian hadith scholar, Quran exegete, and historian. He authored Al-Bidaya wa'l-Nihaya (The Beginning and the End), a seminal work widely regarded as a key reference in the Islamic world. (TR: *İbn Kesîr (1301 - 1373), Suriyeli muhaddis, müfessir ve tarihçi. İslam dünyasında kaynak bir eser olan El Bidaye ve'n Nihaye'yi yazmıştır.*)	What is Ibn Kathir's nationality? (TR: *İbn Kesir'in uyruğu nedir?*)	Syrian (TR: *Suriyeli*)	Syrian (TR: *Suriyeli*)
The structure used as an observatory was built between 1934 and 1936, based on the designs of architect Arif Hikmet Holtay. Architecturally, it is considered part of the rationalist-modernist movement. It is still actively used for the same purpose today. (TR: *Gözlemevi olarak kullanılan yapı, 1934-1936 yılları arasında, mimar Arif Hikmet Holtay'ın çizimlerine göre inşa edilmiştir. Yapı, mimari olarak rasyonel-modernist akım içinde değerlendirilmektedir. Hâlen etkin olarak aynı amaçla kullanılmaktadır.*)	In which years was the observatory built? (TR: *Gözlemevi hangi yıllar arasında inşa edilmiştir?*)	1934-1936	1934-1936
The Treaty of Vasvar is a peace treaty. It was signed on August 18, 1618, between the Ottoman Empire and Poland. The treaty was signed in the city of Vasvar. (TR: *Vasvar Antlaşması bir barış antlaşmasıdır. Vasvar Antlaşması 18 Ağustos 1618 tarihinde imzalanmıştır. Vasvar Antlaşması, Osmanlı Devleti ile Lehistan arasında imzalanmıştır.Vasvar Antlaşması Vasvar kentinde imzalanmıştır.*)	The Treaty of Vasvar was signed between the Ottoman Empire and which state? (TR: *Vasvar Antlaşması, Osmanlı Devleti ile hangi devlet arasında imzalanmıştır?*)	Poland (TR: *Lehistan*)	Poland (TR: *Lehistan*)

Context	Question	Ground Truth	Answer Generated by BERTurk
Samarqand died in 1222 during the Mongol invasion in the city of Herat, Afghanistan. Although little is known about his life, he was a prolific medical writer and an interpreter of medical ideas.	In which country did Semerkandi lose his life?	Afghanistan	Afghanistan
(TR: *Semerkandi, 1222 yılında Moğal saldırısı sırasında Afganistan'ın Herat şehrinde öldü. Hayatının az kısmı bilinmesine rağmen o, üretken tıbbi yazar ve tıbbi fikirlerin yorumcusuydu.*)	(TR: *Semerkandi hayatını hangi ülkede kaybetmiştir?*)	(TR: *Afganistan*)	(TR: *Afganistan*)
The Treaty of Vasvar is a peace treaty. It was signed on August 18, 1618, between the Ottoman Empire and Poland. The Treaty of Vasvar was signed in the city of Vasvar.	Where was the Treaty of Vasvar signed?	In the city of Vasvar	In the city of Vasvar
(TR: *Vasvar Antlaşması bir barış antlaşmasıdır. Vasvar Antlaşması 18 Ağustos 1618 tarihinde imzalanmıştır. Vasvar Antlaşması, Osmanlı Devleti ile Lehistan arasında imzalanmıştır.Vasvar Antlaşması Vasvar kentinde imzalanmıştır.*)	(TR: *Vasvar Antlaşması nerede imzalanmıştır?*)	(TR: *Vasvar kentinde*)	(TR: *Vasvar kentinde*)

To provide a more fine-grained analysis of the models' behaviors, we categorized the test questions into semantic types as follows: *Date*, *Definition*, *Location*, *Numeric*, *Person*, and *Other*. Table 7 presents a comparative evaluation of each model's EM and BLEU scores across these question types. This breakdown allows for a deeper understanding of the specific strengths and weaknesses of each model, particularly in relation to their ability to handle factual, numerical, or descriptive content. As shown in Table 7, *BERTurk* performs relatively well across all categories, while *T5-Small* struggles especially with entity- and number-based queries.

Table 7. Performance Comparison of the Five Transformer Models—*BERTurk*, *DistilBERT*, *ELECTRA*, *T5-Small*, and *XLM-RoBERTa*—Based on Question Type. The Results Are Reported in Terms of EM And BLEU Score for Each Semantic Category: *Date*, *Definition*, *Location*, *Numeric*, *Person*, And *Other*. This Categorization Enables the Evaluation of Model Effectiveness in Handling Different Forms of Factual and Descriptive Information.

	EM					BLEU Score				
Question Type	***BERTurk***	***DistilBERT***	***ELECTRA-Turkish***	***T5-Small***	***XLM-RoBERTa***	***BERTurk***	***DistilBERT***	***ELECTRA-Turkish***	***T5-Small***	***XLM-RoBERTa***
Date	**0.581**	0.393	0.579	0.018	0.541	**0.248**	0.181	0.262	0.025	0.240
Definition	**0.421**	0.168	0.404	0.017	0.472	**0.354**	0.166	0.319	0.033	0.343
Location	**0.490**	0.303	0.485	0.019	0.545	0.290	0.193	**0.300**	0.034	0.313
Numeric	0.288	0.208	**0.296**	0.024	**0.296**	**0.086**	0.082	0.081	0.025	0.062
Person	0.567	0.286	0.607	0.006	**0.632**	0.347	0.225	0.358	0.045	**0.373**
Other	0.397	0.209	**0.440**	0.015	0.430	**0.338**	0.212	0.324	0.031	0.333

Error Analysis

We conducted a detailed analysis of its incorrect predictions on the test set to better understand the failure modes of the fine-tuned *BERTurk* model. From the error cases, several recurring patterns emerged:

- *Surface Form Mismatches.* Many predictions were semantically correct but formatted differently (e.g., "*1258*" vs. "*1258 yılında*" [EN: "*In the year of 1258*"]). These indicate that the model understands the context but adds temporal or locational suffixes common in Turkish, resulting in technically mismatched spans.
- *Entity Confusion.* Confusion between similar historical figures was observed (e.g., "*Orhan Gazi*" vs. "*Osman Bey*"), especially when both were mentioned in the same context. This points to limitations in entity disambiguation.
- *Lexical Variants & Transliteration.* Errors stem from the presence of abbreviations and orthographic variations, especially for named entities.
- *Scope Errors in Long Contexts.* In multi-sentence contexts, the model occasionally selected the wrong sub-span that partially included the correct information.

These findings suggest that while the model captures the semantic neighborhood of the correct answers quite well, it struggles with token span exactness, morphological variants, and entity resolution in crowded contexts.

Web GUI of the Turkish QA System

To facilitate real-time interaction with the developed Turkish QA model, a web-based application was implemented using *Streamlit* ("Streamlit: A Faster Way to Build and Share Data Apps," 2025), a lightweight and efficient open-source framework for deploying ML models. This application provides an interactive graphical user interface (GUI) that enables users to input a textual context and a corresponding question, allowing the system to extract and present the most relevant answer. The backend employs the best-performing model, namely *BERTurk*. The model processes the input by first applying WordPiece tokenization, ensuring compatibility with the transformer architecture. The tokenized text is then fed into the model, which predicts the start and end indices of the most probable answer span within the given context. The extracted tokens are subsequently reconstructed into human-readable text and displayed in the interface. If no valid answer is found, an appropriate message is returned to inform the user.

The *Streamlit*-based web UI was designed to be intuitive and computationally efficient. The interface consists of two primary input fields—one for entering the context paragraph and another for posing a question—as well as a "*Get Answer*" button that triggers the inference process. Upon submission, the system tokenizes the input, performs inference using the transformer model, and displays the extracted answer in real time. To optimize performance, *Streamlit*'s internal caching mechanism was employed to ensure that the model is loaded only once per session, significantly reducing inference time. The backend processing follows a modular structure, consisting of preprocessing, inference, and post-processing modules. The preprocessing module tokenizes and encodes the input, the inference module computes the most likely answer span, and the post-processing module reconstructs the extracted tokens into a coherent answer. The screenshot in Fig. 6 illustrates the developed interactive web GUI of the proposed Turkish QA system, where users can input a context and question to retrieve an answer from the trained model.

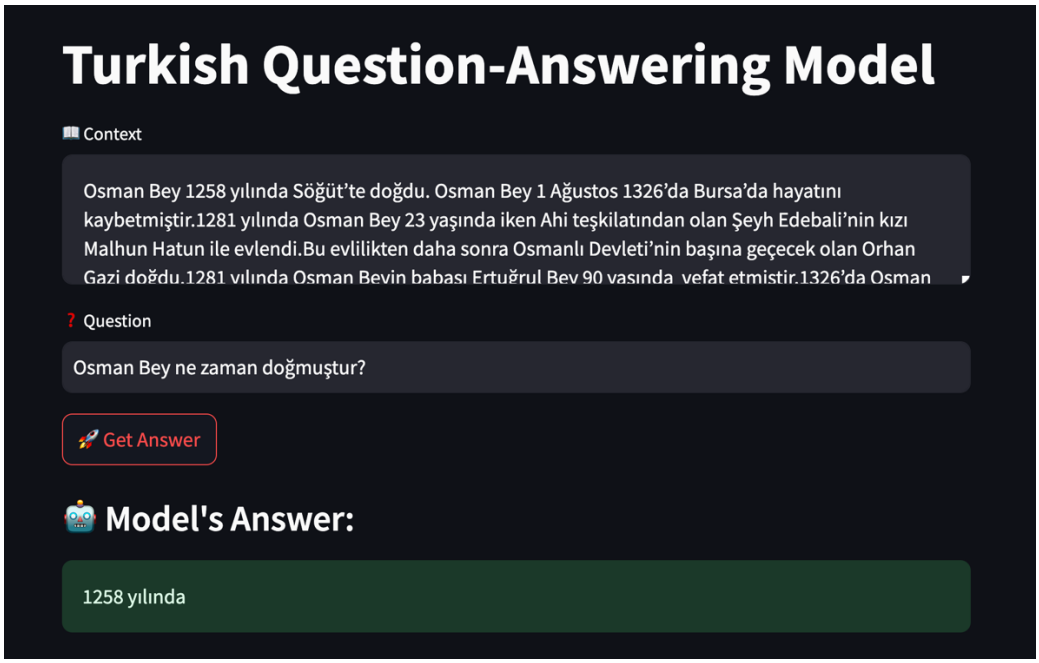

Figure 6. Screenshot Of the Developed *Streamlit*-Based Web Interface for The Proposed Turkish QA System. This Interactive GUI Enables Real-Time Evaluation by Allowing Users to Input a Context and A Question, And Instantly Receive the Model's Extracted Answer. It Highlights the Practical Applicability and User-Friendliness of The System, Facilitating Broader Accessibility and Deployment for Native Turkish Speakers.

Threats to Validity and Limitations

Despite the strong performance of the evaluated transformer models and the methodological rigor applied in this study, several limitations and potential threats to validity exist:

- *Dataset Dependency*. The findings are based on a single Turkish QA dataset (Soygazi et al., 2021). Although it is a gold-standard dataset, the generalizability of results to other Turkish QA datasets or domains (e.g., biomedical or legal texts) remains unverified.
- *Computational Constraints.* Due to hardware limitations, large-scale hyperparameter tuning or multi-fold cross-validation was not feasible. Therefore, the models were evaluated using a fixed training-testing split provided by the dataset.
- *Reproducibility*. Although efforts were made to ensure reproducibility (e.g., fixed random seeds, shared code), slight variations may still occur due to non-deterministic operations in deep learning frameworks.

CONCLUSION

LLM chatbots have become one of the most commonly used – if not the most – ways for people to acquire knowledge on any topic they wish to learn about. This widespread adoption is driven by their ability to provide instant, context-aware responses tailored to individual queries. As a result, they are increasingly integrated into educational platforms, professional environments, and customer support systems. It is safe to say that they have significantly transformed the way people interact with software in their daily activities. Many applications now offer built-in LLM support, either by integrating their models or leveraging existing ones. There are many studies investigating the effectiveness of LLMs in English QA; however, when it comes to Turkish, the number of studies is very limited, and the existing ones are not comprehensive. To address this limitation, we proposed this study as a comprehensive evaluation of several state-of-the-art Transformer-based LLMs on a gold-standard Turkish QA dataset, aiming to address the underexplored performance of these models in low-resource and morphologically rich languages. The experimental results demonstrated that the *BERTurk* model, which is specifically pre-trained on Turkish text, outperformed other models, including multilingual and distilled variants, in terms of F1-score, EM, and BLEU score despite slightly higher training and inference costs. This superior performance can be attributed to its Turkish-specific pre-training, full-sized architecture, and robust MLM approach, which enable it to better capture the morphological richness and agglutinative structure of Turkish. In contrast, models like *XLM-RoBERTa* and *DistilBERT*, while efficient, showed limitations in handling the nuances of Turkish due to their multilingual training and lighter architectures. The study also highlighted the trade-offs between model performance and computational efficiency, showing that while *DistilBERT* and *T5-Small* offer faster inference and training times along with lower memory consumption, they do so at the expense of reduced accuracy. These findings underscore the importance of developing and fine-tuning language-specific models for low-resource languages to achieve optimal performance in NLP tasks. To support reproducibility and encourage further research, all code used in this study has been made publicly available at: https://github.com/talhakabakus/turkish-qa-transformers.

This study has demonstrated that pre-trained transformer models can be effectively employed for Turkish QA tasks. However, there are several promising directions for future research. First, expanding the availability of high-quality, diverse datasets—particularly from specialized domains such as healthcare, law, and education—will be essential for evaluating the generalizability of QA models across various real-world contexts. Additionally, developing advanced pre-training techniques tailored to the unique morphological and syntactic characteristics of Turkish and other underrepresented languages can further enhance the effectiveness of these models. Future work may also involve extending the current benchmark by incorporating multilingual and cross-lingual QA models, experimenting with zero-shot and few-shot learning settings, and integrating hybrid neural-symbolic reasoning to improve model interpretability. Exploring lightweight, resource-efficient transformer architectures is another important direction, especially for deployment in low-resource or mobile environments. To address some of the limitations identified in this study, future research could incorporate broader datasets, more robust validation strategies such as k-fold cross-validation, and extensive hyperparameter optimization. Finally, leveraging alternative tokenization strategies and applying data augmentation techniques suited to agglutinative languages like Turkish could yield further performance gains. These advancements will be crucial not only for improving QA systems' accuracy and efficiency but also for ensuring their applicability in diverse linguistic and cultural contexts.

REFERENCES

Bilgin, M., Bozdemir, M., and Demir, E. (2024). Performance Analysis of Large Language Models on Turkish Question-Answer Texts. *Proceedings of the 2024 Electrical-Electronics and Biomedical Engineering Conference (ELECO 2024)*, 1–5. Bursa, Türkiye: IEEE. https://doi.org/10.1109/ELECO64362.2024.10847201

Bonov, P. (2025). DeepSeek climbs to top spot of the App Store, beats ChatGPT in the process. Retrieved February6, 2025, from GSMArena website: https://www.gsmarena.com/deepseek_climbs_to_top_spot_of_the_app_store_beats_chatgpt_in_the_process-news-66286.php

Celebi, E., Gunel, B., and Sen, B. (2011). Automatic Question Answering for Turkish with Pattern Parsing. *Proceedings of the 2011 International Symposium on INnovations in Intelligent SysTems and Applications*, 389–393. Istanbul, Türkiye: IEEE. https://doi.org/10.1109/INISTA.2011.5946098

Clark, K., Luong, M. T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*, 1–18. Online.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., … Stoyanov, V. (2020). Unsupervised Cross-lingual Representation Learning at Scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. Online: ACL. https://doi.org/10.18653/v1/2020.acl-main.747

Datasets. (2025). Retrieved April 7, 2025, from Hugging Face website: https://huggingface.co/docs/datasets/en/index

Derici, C., Çelik, K., Kutbay, E., Aydın, Y., Güngör, T., Özgür, A., and Kartal, G. (2015). Question Analysis for a Closed Domain Question Answering System. *Proceedings of the 16th International Conference Computational Linguistics and Intelligent Text Processing (CICLing 2015)*, *9042*. Cairo, Egypt: Springer. https://doi.org/10.1007/978-3-319-18117-2_35

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2019)*, *1*. Minneapolis, Minnesota, USA: ACL.

Evaluate. (2025). Retrieved April 9, 2025, from Hugging Face website: https://huggingface.co/docs/evaluate/index

Hu, K. (2023). ChatGPT sets record for fastest-growing user base - analyst note. Retrieved February 13, 2025, from Reuters website: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science and Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Kotstein, S., and Decker, C. (2024). RESTBERTa: a Transformer-based question answering approach for semantic search in Web API documentation. *Cluster Computing*, *27*(4). https://doi.org/10.1007/s10586-023-04237-x

Kuligowska, K., and Kowalczuk, B. (2021). Pseudo-labeling with transformers for improving Question Answering systems. *Procedia Computer Science*, *192*, 1162–1169. https://doi.org/10.1016/J.PROCS.2021.08.119

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Allen, P. G. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, *1907.11692*, 1–13.

Loshchilov, I., and Hutter, F. (2019). Decoupled Weight Decay Regularization. *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, 1–19. New Orleans, LA, USA.

Luo, K., Lin, F., Luo, X., and Zhu, K. Q. (2018). Knowledge base question answering via encoding of complex query graphs. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2185–2194. https://doi.org/10.18653/v1/d18-1242

MDZ Digital Library team. (2024a). dbmdz/bert-base-turkish-cased. Retrieved January 21, 2025, from Hugging Face website: https://huggingface.co/dbmdz/bert-base-turkish-cased

MDZ Digital Library team. (2024b). dbmdz/electra-base-turkish-cased-discriminator. Retrieved January 17, 2025, from Hugging Face website: https://huggingface.co/dbmdz/electra-base-turkish-cased-discriminator

Mehta, I. (2025). DeepSeek reaches No. 1 on US Play Store | TechCrunch. Retrieved February 6, 2025, from TechCrunch website: https://techcrunch.com/2025/01/28/deepseek-reaches-no-1-on-us-play-store/

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 311–318. Philadelphia, Pennsylvania, USA: ACL.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Proceedings of the Thirty-Third Conference on Neural Information Processing Systems (NIPS 2019)*, 8026–8037. Vancouver, BC, Canada.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., … Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*, 1–67.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (NeurIPS 2019)*, 1–5. Vancouver, BC, Canada.

Soygazi, F., Ciftci, O., Kok, U., and Cengiz, S. (2021). THQuAD: Turkish Historic Question Answering Dataset for Reading Comprehension. *Proceedings of the 6th International Conference on Computer Science and Engineering (UBMK 2021)*. Ankara, Türkiye: IEEE. https://doi.org/10.1109/UBMK52708.2021.9559013

Streamlit: A faster way to build and share data apps. (2025). Retrieved March 2, 2025, from Snowflake Inc. website: https://streamlit.io

The pandas development team. (2020). pandas: Python Data Analysis Library. Retrieved January 7, 2024, from https://pandas.pydata.org

Vazrala, S., and Khatoon Mohammed, T. (2025). RBTM: A Hybrid gradient Regression-Based transformer model for biomedical question answering. *Biomedical Signal Processing and Control*, *102*, 107325. https://doi.org/10.1016/J.BSPC.2024.107325

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 1–4. https://doi.org/10.21105/joss.03021

Wolf, T., Debut, L., Sanh, V., Chaumond, J., and ... (2020). Transformers: State-of-the-art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, *1910.03771*, 38–45. Online: ACL.

Xu, K., Reddy, S., Feng, Y., Huang, S., and Zhao, D. (2016). Question answering on freebase via relation extraction and textual evidence. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, *4*, 2326–2336. Berlin, Germany: ACL. https://doi.org/10.18653/v1/p16-1220

Xue, X., Zhang, J., and Chen, Y. (2024). Question-answering framework for building codes using fine-tuned and distilled pre-trained transformer models. *Automation in Construction*, *168*, 105730. https://doi.org/10.1016/J.AUTCON.2024.105730

Yu, M., Yin, W., Hasan, K. S., dos Santos, C., Xiang, B., and Zhou, B. (2017). Improved neural relation detection for knowledge base question answering. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (ACL 2017)*, *1*, 571–581. Vancouver, Canada: ACL. https://doi.org/10.18653/v1/P17-1053

Zhu, S., Cheng, X., and Su, S. (2020). Knowledge-based question answering by tree-to-sequence learning. *Neurocomputing*, *372*, 64–72. https://doi.org/10.1016/j.neucom.2019.09.003