Journal on
Artificial Intelligence

Tech Science Press

https://doi.org/10.32604/jai.2026.075857

ARTICLE

Fine Tuned QA Models for Java Programming

Jeevan Pralhad Tonde* **and Satish Sankaye**

Dr. G.Y. Pathrikar College of CS and IT, MGM University, Chhatrapati Sambhaji Nagar, India

*Corresponding Author: Jeevan Pralhad Tonde. Email: jptonde@gmail.com

Received: 10 November 2025; Accepted: 14 January 2026; Published: 13 February 2026

ABSTRACT: As education continues to evolve alongside artificial intelligence, there is growing interest in how large language models (LLMs) can support more personalized and intelligent learning experiences. This study focuses on building a domain-specific question answering (QA) system tailored to computer science education, with a particular emphasis on Java programming. While transformer-based models such as BERT, RoBERTa, and DistilBERT have demonstrated strong performance on general-purpose datasets like SQuAD, they often struggle with technical educational content where annotated data is scarce. To address this challenge, we developed a custom dataset, JavaFactoidQA, consisting of 1000 fact-based question–answer pairs derived from Java course materials and textbooks. A two-step fine-tuning strategy was adopted, in which models were first fine-tuned on the SQuAD dataset to capture general language understanding and subsequently fine-tuned on the Java-specific dataset to adapt to programming terminology and structure. Experimental results show that RoBERTa-Base achieved the best performance, with an F1 score of 88.7% and an Exact Match (EM) score of 82.4%, followed closely by BERT-Base and DistilBERT. The results were further compared with domain-specific QA models from healthcare and finance, demonstrating that the proposed approach performs competitively despite using a relatively small dataset. Overall, this study shows that careful dataset design combined with sequential fine-tuning enables effective adaptation of transformer-based QA models for educational applications, including automated assessment, intelligent tutoring, and interactive learning environments. Future work will explore extending the approach to additional subjects, incorporating cognitive-level tagging, and evaluating performance on broader educational QA benchmarks.

KEYWORDS: Question answering; transfer learning; factoid question finetuning; large language model; transformers; BERT

1 Introduction

In today's rapidly evolving educational landscape, artificial intelligence is playing an increasingly central role in reshaping teaching and learning practices. One of the most promising developments in this area is the use of large language models (LLMs) [1] such as BERT [2], RoBERTa [3], and DistilBERT [4] for building intelligent question answering (QA) systems. These models have demonstrated strong performance on benchmark datasets such as SQuAD [5], highlighting their ability to understand and respond to complex natural language queries.

Despite their success in general domains, applying these models to specialized areas such as Java programming remains challenging. Educational content in computer science is often characterized by technical terminology, structured explanations, and strong conceptual dependencies, while well-annotated domain-specific datasets are scarce. This gap limits the effectiveness of existing QA systems in supporting learners in programming-related subjects, where accuracy and clarity are essential.

To address this challenge, we adopt a sequential fine-tuning strategy in which models are first fine-tuned on a general-purpose dataset such as SQuAD to learn broad language patterns and question–context alignment, and are then further fine-tuned on a smaller domain-specific dataset to specialize in a particular field. This approach has shown promising results in other domains including healthcare and finance, where annotated data is limited but domain accuracy is critical [6–8].

In this study, we apply sequential fine-tuning [9,10] to three widely used transformer models, DistilBERT, BERT-Base, and RoBERTa-Base [3,11,12], to develop a QA system tailored for Java programming education. To support this process, we construct a custom dataset named JavaFactoidQA [13], consisting of 1000 fact-based question-answer pairs derived from Java textbooks and lecture materials. The dataset was annotated using the Haystack annotation tool [14] to ensure precise alignment between questions, contexts, and answer spans.

Beyond technical performance, this work is motivated by its potential educational impact. A well-adapted QA system can serve not only as a question-answering tool, but also as an intelligent tutor, support personalized assessments, and assist both students and instructors in retrieving relevant information efficiently. By adapting LLMs [7] to Java programming content, this study aims to bridge the gap between powerful language models and practical classroom needs.

Through this research, we demonstrate that even with limited computational and annotation resources, it is possible to build effective domain-specific QA systems that enhance learning, support educators, and improve accessibility in programming education.

The primary contributions of this work are summarized as follows:

- We design an extractive question-answering framework that provides accurate answers from given educational content using transformer-based language models.
- We create a domain-specific dataset, JavaFactoidQA, consisting of 1000 manually curated fact-based question-answer pairs derived from Java programming textbooks and lecture materials.
- We adopt a two-stage sequential fine-tuning strategy in which models are first fine-tuned on the general-purpose SQuAD dataset and subsequently adapted to the Java domain, improving performance in a low-resource educational setting.
- We conduct a comparative evaluation of BERT, RoBERTa, and DistilBERT models using standard QA metrics, demonstrating that strong performance can be achieved without specialized architectures or large-scale domain-specific corpora.

Despite the success of transformer-based QA models, many existing domain-specific approaches rely on large-scale domain corpora, specialized architectures, or extensive task-specific modifications. Such requirements limit their applicability in educational environments where annotated data and computational resources are constrained. The innovation of this work lies in demonstrating that a carefully curated small-scale educational dataset combined with a simple two-stage sequential fine-tuning strategy is sufficient to achieve competitive performance. By leveraging standard transformer-based QA models without architectural changes, the proposed approach directly addresses challenges related to data scarcity and resource limitations, making it practical for real-world educational applications.

2 Related Work

The rapid advancement of transformer-based language models like BERT, RoBERTa, and DistilBERT has transformed the landscape of question answering (QA). Initially trained on vast text corpora, these models have been fine-tuned on benchmark datasets such as SQuAD achieving state-of-the-art results and setting a high standard for extractive QA tasks.

Early contributions in this field, such as the work by Li and Pollett [15], demonstrated how a well-structured dataset like SQuAD could guide the development of end-to-end neural QA systems. Their research paved the way for broader adoption of fine-tuning techniques in QA pipelines.

To better understand the evolution of QA systems, Wang [16], Farea and Emmert-Streib [10] provided an extensive survey that reviewed modern datasets and benchmark strategies. This overview clarified how data structure and evaluation methodology impact the performance and reliability of QA models.

As QA systems moved from general domains to more specialized applications, the need for domain-aware fine-tuning became more evident. Guo et al. [9] explored fine-tuning under limited annotation budgets, proposing strategies like pseudo-labeling and dual-phase training to boost model accuracy in domains such as finance. Similarly, in the medical domain, Anisuzzaman et al. [7] applied BERT models to clinical records, achieving strong results by carefully adapting general-purpose models to specialized datasets.

Jeong [17] expanded on this by examining fine-tuning techniques for domain-specific LLMs in legal and multilingual contexts. His findings reinforced that thoughtful pretraining and adaptation strategies can significantly enhance performance across diverse domains.

A different perspective was offered by Lalor et al. [18], who experimented with soft label memorization to improve generalization in QA and natural language inference. This technique was especially beneficial when training data was limited or uncertain.

Practical applications have also gained momentum. Sharma et al. [19] demonstrated how extractive QA models can be successfully fine-tuned for clinical document analysis, employing robust preprocessing and context handling strategies many of which are mirrored in our work on Java programming content.

In educational contexts, Bhattacharyya [20] emphasized the potential of transformer models in supporting intelligent knowledge retrieval. His study pointed out how well-adapted QA systems can serve as powerful tools in education, particularly for STEM and technical subjects.

Domain-specific datasets continue to play a critical role. For instance, Rachmawati and Yulianti [21] introduced StatMetaQA, a QA dataset for Indonesian statistical metadata, showing the growing global interest in building localized, purpose-driven QA systems. Likewise,

Together, these studies establish a strong foundation for domain-specific QA research. They validate our approach of sequential fine-tuning starting with general-domain pretraining and gradually adapting to a focused educational domain like Java programming. Our work builds on these ideas by providing a well-structured dataset and evaluating three popular transformer models in the context of computer science education.

3 Methodology

3.1 Dataset

The JavaFactoidQA dataset was constructed through a structured and manual curation process to ensure domain relevance and annotation quality. Educational content was collected from standard Java programming textbooks and undergraduate-level lecture notes, focusing on core topics such as object-oriented concepts, exception handling, multithreading, inheritance, and interfaces. Each selected paragraph was treated as an independent context passage for question generation.

For each context passage, a fact-based question was manually formulated to assess specific conceptual knowledge present in the text. Answer spans were explicitly annotated by identifying the exact text segment

within the context along with its starting character index, following the SQuAD v1.1 extractive question-answering format. This ensured that each question could be answered strictly from the provided context and that answers were continuous text spans suitable for extractive QA models.

The annotation process was carried out using the Haystack annotation tool, which provides a visual interface for accurately aligning questions, contexts, and answer spans. Each question–answer pair was manually reviewed to verify correctness, clarity, and consistency. Ambiguous or low-quality samples were removed during validation to maintain dataset reliability.

The finalized dataset consists of 1000 high-quality question–answer pairs, which were split into 800 samples for training and 200 samples for validation, as summarized in Table 1.

Table 1: Dataset breakdown.

Number of QA Pairs	Split
800	Training
200	Testing
1000	**Total**

A sample entry from the dataset illustrates how each instance contains a context passage, a fact-based question, and the corresponding answer span with its starting position. This structured representation enables the model to effectively learn how to locate and extract accurate answers from technical educational content. As shown in Fig. 1, each dataset instance is organized into context field, a question field, and an answer annotation containing the exact span and its starting index.


```
{
  "version": "1.0",
  "data": [
    {
      "title": "JavaConceptFactoid",
      "paragraphs": [
        {
          "context": "An exception in Java is an abnormal condition that disrupts the normal program flow. I
          "qas": [
            {
              "id": "f1",
              "question": "What are the five keywords used in Java for exception handling?",
              "answers": [
                {
                  "text": "try, catch, throw, throws, and finally",
                  "answer_start": 105
                }
              ],
              "is_impossible": false
            }
          ]
        }
      ]
    }
  ]
}
```


Figure 1: Sample entry from JavaFactoidQA dataset.

Overall, JavaFactoidQA provides a clean and focused benchmark for training and evaluating extractive question-answering models in the educational domain. The dataset can also serve as a foundation for future

extensions, such as expansion to other programming languages or categorization of questions based on cognitive levels, for example Bloom's Taxonomy [22].

3.2 Model Selection

To evaluate the effectiveness of transformer-based architectures for domain-specific question answering, we selected three widely used pre-trained language models as the foundation for our experiments: DistilBERT, BERT-Base, and RoBERTa-Base [2]. These models were chosen due to their strong performance on extractive QA benchmarks, availability of open-source implementations, and compatibility with the SQuAD dataset format.

DistilBERT is a compact and efficient version of BERT that retains approximately 97% of BERT's language understanding capabilities while significantly reducing model size and inference time. This makes it suitable for low-resource environments or scenarios where computational efficiency is a priority.

BERT-Base (uncased) is a standard 12-layer transformer model pre-trained on large-scale text corpora, including BookCorpus and English Wikipedia. It serves as a strong baseline for a wide range of natural language processing tasks, including extractive question answering, and has been extensively evaluated on benchmark datasets such as SQuAD.

RoBERTa-Base is an optimized variant of BERT that employs improved pre-training strategies, such as dynamic masking and training on larger datasets. These enhancements enable RoBERTa to achieve superior performance across several downstream NLP tasks, including question answering, particularly in scenarios requiring deeper contextual understanding.

All three models were obtained from the Hugging Face Transformers library and were fine-tuned using a consistent two-stage sequential strategy. Each model was first fine-tuned on the general-purpose SQuAD v1.1 dataset and subsequently fine-tuned on the proposed JavaFactoidQA dataset. This model selection and training strategy enabled a comparative analysis across lightweight, standard, and optimized transformer architectures, while assessing their adaptability to technical content in programming education.

3.3 Fine-Tuning and Inference Workflow

To adapt transformer-based language models for domain-specific question answering in Java programming, we employed a two-step sequential fine-tuning approach. This strategy enables the models to first learn general question answering patterns and then specialize in domain-specific technical content.

In the first stage, the pre-trained models, namely DistilBERT, BERT-Base, and RoBERTa-Base, were fine-tuned on the SQuAD v1.1 dataset. This stage allows the models to learn generic language understanding, question–context alignment, and answer span prediction. During training, the models learn to identify the start and end positions of answers within a given context using large-scale and diverse question–answer examples. Fine-tuning on SQuAD helps stabilize training and improves performance on extractive question answering tasks.

In the second stage, the SQuAD-fine-tuned models were further fine-tuned on the proposed JavaFactoidQA dataset. This step adapts the learned representations to Java-specific terminology, programming concepts, and structured educational explanations. The extractive QA formulation is retained across both stages to ensure consistency. Initializing this phase with weights already optimized for question answering enables faster convergence and reduces the risk of overfitting, despite the relatively small size of the domain-specific dataset.

For both fine-tuning stages, each input was formatted as a single token sequence using the following structure:

[CLS] Question [SEP] Context [SEP]

This format enables the model to distinguish between the question and the context while predicting the most probable start and end positions of the answer span within the context. The overall training and inference workflow is illustrated in Fig. 2.

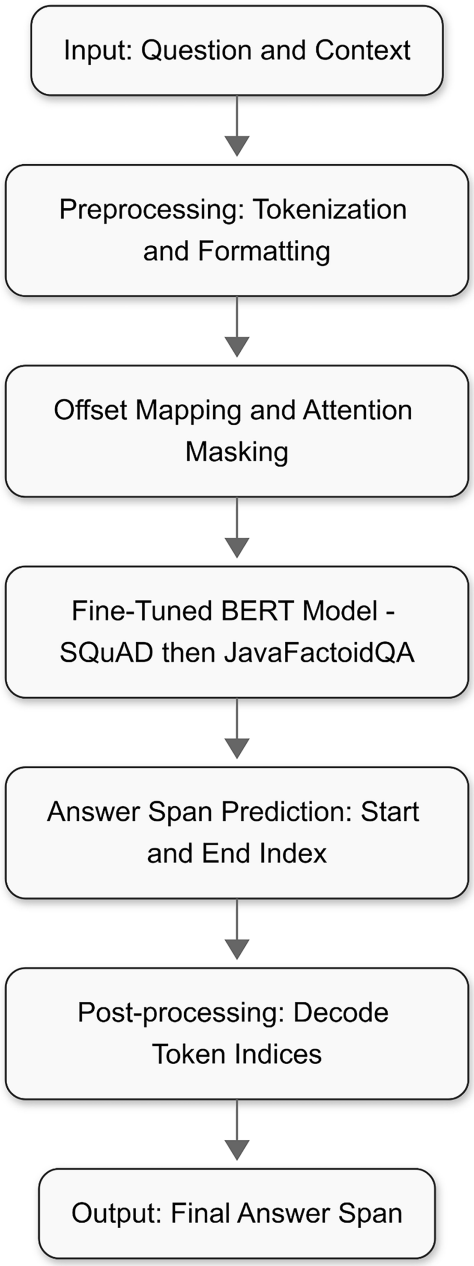

Figure 2: Overall system flow.

Training was performed using the Hugging Face Trainer API with the AdamW optimizer. A learning rate of 2e−5, a batch size of 8, and 2 training epochs were used for each fine-tuning stage. The maximum sequence length was set to 384 tokens, and a sliding window approach with a stride of 128 tokens was applied to handle longer context passages without losing relevant information. Model performance was evaluated using a validation set, and checkpoints were saved after each epoch. Training loss was computed using a cross-entropy objective that measures the difference between predicted and true answer span positions.

By separating general language learning from domain adaptation, the two-step fine-tuning strategy improves answer extraction accuracy and enhances training stability in low-resource educational settings such as Java programming.

3.4 Preprocessing the Data

Before training, a preprocessing step was applied to ensure that the dataset conformed to the input requirements of transformer-based models. Each question–answer instance in the JavaFactoidQA dataset consists of a context paragraph, a corresponding question, and an answer annotated with its exact starting character index within the context.

Model-specific tokenizers, such as Bert-Tokenizer and Roberta-Tokenizer, were used to convert textual input into token sequences. Token-to-character alignment was maintained using offset mappings, which enabled accurate identification of the start and end token positions corresponding to the annotated answer spans.

Since transformer models have a fixed maximum input length, the sequence length was capped at 384 tokens. For longer context passages, a sliding window strategy with an overlap of 128 tokens was employed to ensure that important information was not truncated. Padding was applied to maintain uniform input lengths, and attention masks were used to indicate valid tokens during training. The final preprocessed data was stored using the Hugging Face dataset format to facilitate efficient model training.

3.5 Inference Procedure

During inference, the trained models were provided with a question and a relevant context paragraph, and the objective was to identify the answer span within the given context. The input was formatted in the same manner as during training, as shown below:

[CLS] Question [SEP] Context [SEP]

The model computes probability distributions over token positions and selects the most likely start and end positions for the answer span. When the context exceeded the maximum sequence length, it was divided into overlapping segments using the same sliding window approach applied during preprocessing. Predictions were generated for each segment, and the answer span with the highest overall confidence score was selected.

3.6 Working of the Fine-Tuned BERT-Based QA Model

BERT-based models are effective at understanding contextual relationships between words due to their self-attention mechanism, which allows each token to attend to all other tokens in the input sequence. This capability enables the model to capture both local and global contextual dependencies within a passage.

When a fine-tuned BERT-based QA model receives a question and its corresponding context, it generates contextualized embeddings for each token. A task-specific output layer then predicts the probability of

each token being the start or end of the answer span. These predictions are optimized using a cross-entropy loss function during training, and model parameters are updated through backpropagation.

The combination of large-scale pre-training, domain-specific fine-tuning on Java programming content, and robust contextual modeling enables BERT and its variants to accurately identify answer spans in structured and technical educational texts, such as programming documentation and lecture notes.

4 Results & Discussion

The objective of this study is to evaluate the effectiveness of transformer-based models, namely DistilBERT, BERT-Base, and RoBERTa-Base, for domain-specific question answering in the context of Java programming education. All models were trained using a two-stage sequential fine-tuning strategy, with initial fine-tuning on the general-purpose SQuAD v1.1 dataset followed by domain adaptation on the proposed JavaFactoidQA dataset.

Model performance was evaluated using standard extractive QA metrics, including F1 score and Exact Match (EM). In addition, inference time and model size were considered to analyze the trade-off between efficiency and accuracy.

4.1 Model Performance Analysis

DistilBERT demonstrated fast convergence and competitive performance despite having significantly fewer parameters. It performed well on fact-based and definition-oriented questions but showed limitations when handling questions that required deeper contextual reasoning. As shown in Table 2 DistilBERT achieved an F1 score of 86.3% and an EM score of 80.1%, while offering the fastest inference time of 45 ms per sample.

Table 2: Performance of the model.

Model	Epochs	F1 Score (%) [23,24]	Exact Match EM %) [23,25]	Inference Time (ms/sample)	Model Size (M params)
DistilBERT	2	86.3	80.1	45	66
BERT-Base	2	87.2	81.2	72	110
RoBERTa-Base	2	88.7	82.4	75	125

BERT-Base exhibited improved consistency and stronger contextual understanding compared to DistilBERT. It handled moderately complex queries related to inheritance, polymorphism, and synchronization with higher reliability. According to Table 2 BERT-Base achieved an F1 score of 87.2% and an EM score of 81.2%, confirming its effectiveness as a robust baseline for domain adaptation.

RoBERTa-Base achieved the best overall performance among the evaluated models. It demonstrated superior contextual alignment and robustness when answering complex questions involving subtle distinctions within the context. RoBERTa-Base achieved an F1 score of 88.7% and an EM score of 82.4%, benefiting from its optimized pre-training strategy.

Overall, the trade-off between model size and performance is evident. DistilBERT offers faster inference and lower memory usage, while RoBERTa-Base achieves higher accuracy at the cost of increased computational complexity, as summarized in Table 2.

4.2 Comparison with Domain-Specific QA Approaches

To assess the effectiveness of the proposed approach, the performance of the sequentially fine-tuned RoBERTa-Base model was compared with existing domain-specific QA systems reported in the literature across finance, clinical, and legal domains.

As shown in Table 3, the proposed RoBERTa-Base model achieved an F1 score of 88.7% and an EM score of 82.4% on the Java educational domain dataset, using a consistent two-stage fine-tuning pipeline based on SQuAD and JavaFactoidQA.

Table 3: Performance comparison of RoBERTa-based QA model with existing domain-specific question answering approaches.

Study/Model	Domain	F1 Score (%)	EM (%)	Notes
This Work: RoBERTa-Base	**Java (Edu)**	**88.7**	**82.4**	**2-stage fine-tuning on SQuAD + JavaFactoidQA**
Guo et al. (2023)—BERT + Pseudo-Labelling	Finance	84.5	76.3	Low-resource domain-specific QA with pseudo-annotations
Anisuzzaman et al. (2025)—BERT	Clinical QA	86.2	78.9	Annotated electronic health records (EHR)
Jeong 2024 RoBERTa (Korean Legal)	Legal/Korean	87.1	79.5	Multilingual domain model with custom task heads
Lalor et al. (2017) BiLSTM + Soft Labels	QA/NLI	80.4	–	Soft label memorization for general QA

It should be noted that the baseline models listed in Table 3 were not Pre-trained using the SQuAD and JavaFactoidQA datasets. Their reported results are taken directly from the respective publications, each employing different datasets and training strategies. In contrast, all models evaluated in this study followed a consistent two-stage fine-tuning pipeline.

4.3 Discussion and Limitations

The results demonstrate that sequential fine-tuning effectively adapts pre-trained transformer models to domain-specific educational content, even when only a modest-sized dataset is available. The consistent improvements observed across all three models indicate that the proposed training strategy generalizes well across different model sizes and architectures.

It is important to emphasize that the goal of this study is not to introduce architectural innovations, but to evaluate a practical and reproducible training strategy for educational question answering. By leveraging standard transformer-based models and focusing on data quality and training methodology, the proposed approach remains accessible to educational institutions with limited computational and annotation resources.

A limitation of this study is that evaluation was conducted primarily on the JavaFactoidQA dataset. While this dataset is representative of Java programming education, evaluating the proposed pipeline on additional educational QA benchmarks, such as CQuAE, would further strengthen the assessment of

generalizability. Differences in dataset structure and annotation schemes prevented direct inclusion of these benchmarks in the current study, and their evaluation is left for future work.

5 Conclusion

In this work, we investigated how transformer-based language models can be effectively adapted for domain-specific question answering using a sequential fine-tuning approach. By first fine-tuning the models on the general-purpose SQuAD dataset and subsequently adapting them to the custom JavaFactoidQA dataset, models such as DistilBERT, BERT-Base, and RoBERTa-Base achieved strong performance on educational content related to Java programming.

Among the evaluated models, RoBERTa-Base achieved the best overall performance, demonstrating strong contextual understanding and accurate answer extraction, even for complex and nuanced questions. BERT-Base also performed reliably across most question types, while DistilBERT offered faster inference with only a modest reduction in accuracy, making it a practical choice for real-time applications or environments with limited computational resources. Future work will extend evaluation to educational benchmarks such as CQuAE to further validate generalizability

To assess the impact of intermediate fine-tuning on a general-purpose dataset, an ablation study was conducted by comparing models trained directly on the JavaFactoidQA dataset with models trained using a two-stage sequential fine-tuning strategy, namely SQuAD followed by JavaFactoidQA. As shown in Table 4, models incorporating the intermediate SQuAD fine-tuning stage consistently outperformed those trained solely on the domain-specific dataset across all evaluation metrics.

Table 4: Ablation study on the effect of intermediate SQuAD fine-tuning.

Model	Fine-Tuning Strategy	F1 Score (%)	Exact Match (%)
RoBERTa-Base	JavaFactoidQA only	84.9	77.8
RoBERTa-Base	SQuAD JavaFactoidQA	88.7	82.4
BERT-Base	JavaFactoidQA only	83.6	76.5
BERT-Base	SQuAD JavaFactoidQA	87.2	81.2
DistilBERT	JavaFactoidQA only	82.1	74.9
DistilBERT	**SQuAD JavaFactoidQA**	**86.3**	**80.1**

For RoBERTa-Base, intermediate SQuAD fine-tuning resulted in an improvement of 3.8% in F1 score and 4.6% in Exact Match, indicating more accurate answer span prediction and improved contextual alignment. Similar performance gains were observed for BERT-Base and DistilBERT. In addition to these quantitative improvements, models trained without SQuAD fine-tuning exhibited slower convergence and less stable answer boundary detection during training.

These findings demonstrate that intermediate fine-tuning on a large-scale general QA dataset plays a crucial role in stabilizing training and enhancing performance when adapting transformer-based models to low-resource educational domains. The results strongly justify the use of a two-stage sequential fine-tuning pipeline for domain-specific question answering tasks in programming education.

Future work will extend the evaluation of the proposed framework to additional educational benchmarks and advanced domain-specific QA models, including knowledge-aligned and preference-based approaches, to further assess comparative performance and generalizability.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Jeevan Tonde: Conceived the research idea, curated the dataset (JavaFactoidQA), implemented the model fine-tuning, conducted the experiments, analyzed the results, and prepared the manuscript. And Dr. Satish Sankaye: Provided research guidance, supervised the methodology and experimental design, reviewed the manuscript, and offered critical insights to improve the overall quality of the study. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not Applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

LLM	Large Language Model
QA	Question Answering
EM	Exact Match
SQuAD	Stanford Question Answering Dataset
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers.

References

1. Naveed H, Khan AU, Qiu S, Saqib M, Anwar S, Usman M, et al. A comprehensive overview of large language models. ACM Trans Intell Syst Technol. 2025;16(5):106:1–72. doi:10.1145/3744746.
2. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T, editors. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2019 Jun 2–7; Minneapolis, MN, USA. Volume 1 (Long and Short Papers). Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 4171–86. doi:10.18653/v1/N19-1423.
3. Comparative study of bert models and roberta in transformer based question answering | IEEE Conference Publication | IEEE Xplore [Online]. [cited 2025 Dec 30]. Available from: https://ieeexplore.ieee.org/abstract/document/10205622.
4. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108. 2020. doi:10.48550/arXiv.1910.01108.
5. Rajpurkar P, Zhang J, Lopyrev K, Liang P. SQuAD: 100,000+ questions for machine comprehension of text. In: Su J, Duh K, Carreras X, editors. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing; 2016 Nov 1–5; Austin, TX, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2016. p. 2383–92. doi:10.18653/v1/D16-1264.
6. Yadav H, Yadav P, Yadav N, Chaudhary P. AI in healthcare: a survey on medical question answering system. South East Eur J Public Health. 2024;XXIV:1287–98. doi:10.70135/seejph.vi.2683.
7. Anisuzzaman DM, Malins JG, Friedman PA, Attia ZI. Fine-tuning large language models for specialized use cases. Mayo Clin Proc Digit Health. 2025;3(1):100184. doi:10.1016/j.mcpdig.2024.11.005.
8. Fatemi S, Hu Y. Enhancing financial question answering with a multi-agent reflection framework. In: Proceedings of the 5th ACM International Conference on AI in Finance; 2024 Nov 14–17; Brooklyn, NY, USA. New York, NY, USA: ACM; 2024. p. 530–7. doi:10.1145/3677052.3698686.
9. Guo K, Diefenbach D, Gourru A, Gravier C. Fine-tuning strategies for domain specific question answering under low annotation budget constraints. In: Proceedings of the 2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI); 2023 Nov 6–8; Atlanta, GA, USA. p. 166–71. doi:10.1109/ictai59109.2023.00032.

10. Farea A, Emmert-Streib F. Understanding question-answering systems: evolution, applications, trends, and challenges. Eng Appl Artif Intell. 2025;156:110997. doi:10.1016/j.engappai.2025.110997.
11. Prytula M. Fine-tuning BERT, DistilBERT, XLM-RoBERTa and Ukr-RoBERTa models for sentiment analysis of Ukrainian language reviews. Stucintelekt. 2024;29:85–97. doi:10.15407/jai2024.02.085.
12. Comparative analyses of BERT, RoBERTa, DistilBERT, and XLNet for text-based emotion recognition [Online]. [cited 2025 Aug 6]. Available from: https://www.researchgate.net/publication/346443459_Comparative_Analyses_of_BERT_RoBERTa_DistilBERT_and_XLNet_for_Text-based_Emotion_Recognition.
13. qna/JavaFactoidQA.json at Main jptonde/qna GitHub [Online]. [cited 2025 Aug 24]. Available from: https://github.com/jptonde/qna/blob/main/JavaFactoidQA.json.
14. Haystack | Haystack [Online]. [cited 2025 Dec 1]. Available from: https://haystack.deepset.ai/.
15. Li B, Pollett DC. Question answering system on SQuAD dataset using an end-to-end neural network. [cited 2025 Jan 1]. Available from: https://share.google/0fZxbGAaJ6J06ng1I.
16. Wang Z. Modern question answering datasets and benchmarks: a survey. arXiv:2206.15030. 2022. doi:10.48550/arxiv.2206.15030.
17. Jeong C. Fine-tuning and utilization methods of domain-specific LLMs. J Intell Inf Syst. 2024;30(1):93–120. doi:10.13088/jiis.2024.30.1.093.
18. Lalor JP, Wu H, Yu H. Soft label memorization-generalization for natural language inference. arXiv:1702.08563. 2017. doi:10.48550/arxiv.1702.08563.
19. Sharma A, Feldman DI, Jain A, Feldman, Jain A. Fine-tuning pre-trained extractive QA models for clinical document parsing. arXiv:2312.02314. 2023. doi:10.48550/arXiv.2312.02314.
20. Bhattacharyya A. Revolutionizing knowledge retrieval: a comprehensive study of question answering system. SSRN J. 2023. doi:10.2139/ssrn.4649585.
21. Rachmawati N, Yulianti E. StatMetaQA: a dataset for closed domain question answering in Indonesian statistical metadata. Data Brief. 2024;57:110816. doi:10.1016/j.dib.2024.110816.
22. A revision of bloom's taxonomy: an overview: theory into practice: Vol 41, No 4 [Online]. [cited 2025 Dec 1]. Available from: https://www.tandfonline.com/doi/abs/10.1207/S15430421TIP4104_2.
23. Huang H, Xu H, Wang X, Silamu W. Maximum F1-score discriminative training criterion for automatic mispronunciation detection. IEEE/ACM Trans Audio Speech Lang Process. 2015;23(4):787–97. doi:10.1109/taslp.2015.2409733.
24. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genom. 2020;21(1):6. doi:10.1186/s12864-019-6413-7.
25. Naidu G, Zuva T, Sibanda EM. A review of evaluation metrics in machine learning algorithms. In: Artificial intelligence application in networks and systems. Cham, Switzerland: Springer International Publishing; 2023. p. 15–25. doi:10.1007/978-3-031-35314-7_2.