International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I8P116

Volume 74 Issue 8, 239-248, August 2026

Original Article

Design and Implementation of a Scalable AI-Based Semantic Evaluation System for Hindi Text Using Transformer Models

Nirja D Shah[1*], Jyoti Pareek[2]

[1,2]*Department of Computer Science, Gujarat University, Ahmedabad, Gujarat, India*

[1*]*Corresponding Author : nirjashah@gujaratuniversity.ac.in*

Received: 18 March 2026 Revised: 05 July 2026 Accepted: 22 July 2026 Published: 29 August 2026

Abstract - *Evaluating linguistically diverse descriptive answers in a consistent and accurate manner in modern digital education systems is a growing challenge, especially in low-resource languages like Hindi. Traditional lexical and rule-based grading systems cannot adequately reflect the meaning behind the words, negation, paraphrasing, and so on, which leads to low grading reliability. To overcome these limitations, this study proposes an automated evaluation framework with intelligent rule-based linguistic preprocessing and transformer-based deep learning. The framework uses a fine-tuned multilingual BERT (mBERT) model bhavikardeshna/multilingual-bert-base-cased-hindi to provide contextual embeddings, and cosine similarity-based semantic alignment with the model l3cube-pune/hindi-sentence-similarity-sbert is used to provide automated scores. With optimal setting of learning rate = 5×10^{-4}, batch size = 24 and epochs = 40, accuracy, precision, recall and F1 score of 78.9%, 80.6%, 77.4% and 79.0% respectively is achieved on HindiRC-Data-master dataset (24 passages, 127 question-answer pairs, grades 2-5) which is more than 14% higher than lexical similarity baselines and is better than previous Hindi QA architectures without domain-specific preprocessing pipelines. The suggested system will save about 40% manual grading, and will enable scalable, consistent and repeatable assessment.*

Keywords - *Automated answer evaluation, Educational assessment, Hindi Natural Language Processing, Intelligent automation, Semantic similarity, Transformer-based models.*

1. Introduction

Every year, millions of students across India sit for examinations in Hindi, their mother tongue, the language of their thoughts, their stories, and their learning. But when it comes to assessing what those students have written, it still stubbornly requires manual effort, is extremely time-consuming and is inherently inconsistent. Two teachers within the same government school, reading the same answer, might give different marks to a descriptive answer in Ahmedabad and Lucknow. It's not intentional on the part of the human, it's a structural shortcoming of human grading in large volumes.

India's education system is very large, and the National Council of Educational Research and Training (NCERT) is responsible for the curricula that are used by hundreds of millions of students. The subjectivity and time requirement of descriptive answer marking is at the fore of this scale. Previous studies have shown that the set of scores that are awarded for an identical answer can differ significantly [1]. The enhanced scalability and consistency in automating such workflows through the integration of AI and NLP in educational systems have been shown to be promising. Hindi poses a special language problem where automatic evaluation is much more challenging than well-resourced languages. The language of Hindi is morphologically rich, has complex patterns of negation, and students rarely paraphrase answers literally. Although Hindi is the third most widely used language in the world, research and NLP tooling is very limited, especially for Hindi. Early systems based on lexical overlap measures like BLEU or exact string matching were not able to cope with these phenomena. An important milestone came with BERT [7], which introduces bidirectional contextual dependencies with the use of self-attention. Multilingual BERT does this by cross-lingual pertaining to low-resource languages such as Hindi.

The biggest limitation is that there is no system available that offers a fully automated evaluation pipeline with language awareness for Hindi descriptive answers for primary education. The present work would fill up this gap. The planned model is fine-tuning the bhavikardeshna/multilingual-bert-base-cased-hindi model on HindiRC-Data-master (24 passages, 127 QA pairs, grades 2-5), followed by scoring answers using cosine similarity as a

result of the l3cube-pune/hindi-sentence-similarity-sbert sentence transformer. Contributions:

1) A complete reproducible Hindi evaluation pipeline;
2) Systematic hyperparameter optimization via grid search;
3) Empirical benchmarking achieving 78.9% accuracy; and
4) Transparent analysis of system limitations with a future roadmap.

2. Literature Review

The automated scoring of descriptive responses has come a long way in 60 years. Early LSA-based systems, such as Intelligent Essay Assessor and GradeAid [1], represent semantic similarity based on statistical co-occurrence but fail to capture word order, negation, and contextual inference, which is especially problematic in morphologically rich languages.

Khurana et al. [2] presented a jointly designed Hindi QA and handwritten evaluation system that validated the automation of QA systems in Indian languages, but also noted the issues of linguistic variability and limited annotated corpora.

Singh et al. [3] presented the work of H-AES for Hindi essay scoring, Beseiso et al. [4] introduced the machine learning models to approximate the human judgment, and Abdul Salam et al. [5] presented the effective machine learning models of CNN LSTM architectures for Arabic short answer grading. Ensemble-based classification methods are also found to be effective in classifying the Hindi language, for which the main drawback is the scarcity of resources in the context of the Hindi language specifically. Preprocessing techniques, including tokenization, lemmatization, stop word removal, have been found to have a significant effect on the accuracy of downstream language classifiers, both in general and in language-specific stop word lists for low-resource language classification tasks.

In multilingual scenarios, Gupta et al. [12] achieved high performance when using deep networks for English-Hindi QA; Gupta and Khade [13] developed an ensemble of multilingual BERT for comprehension; and Lahoti et al. [14] used an ensemble of multilingual BERT for English-Hindi-Malayalam. Challenges from translation noise were reported in Tamil RC [10]. Morphological variation and handling negation are highlighted by the recent Hindi NLP surveys as the areas of ongoing challenges, and the present work directly caters to these areas of challenge. Transformer fine-tuning has been shown to be the standard method for performing text understanding tasks when applied to domain-specific tasks such as classification, with BERT fine-tuned on such a task. The contribution of sentence-level embeddings to evaluation has significantly increased. Sentence-BERT (SBERT) is a variant of BERT using siamese and triplet network structures to obtain sentence embeddings that are comparable, in a meaningful sense, using cosine similarity, as introduced by Reimers and Gurevych [15]. Using only one sentence at a time, SBERT is able to reduce the pairwise comparison time from 65 hours to about 5 seconds across 10,000 sentences, all while maintaining competitive accuracy on Semantic Textual Similarity (STS) benchmarks. This is in contrast to standard BERT, which creates an $O(n^2)$ computational overhead for pairwise comparison.

This computational efficiency property is directly exploited in the scoring phase of the proposed framework. For automated short answer grading specifically, Zhu et al. [18] demonstrated that BERT-based deep neural networks achieve significantly higher correlation with human judgment than prior feature-engineered approaches on standard ASAG benchmarks, while Haller et al. [17] provided a systematic survey showing the progression from word embeddings to transformer models in this task. The limitation of this study is the same as that which Sridevi Bonthu [22] showed: that domain-specific pre-training of PLMs significantly improves ASAG performance when labeled training data is limited. Recent studies exploring GPT-4 and LLM-based grading [19, 20, 21] have shown promising results in zero-shot and few-shot settings, though these systems face challenges around explainability, cost, and deployment in low-resource language contexts that the present transformer fine-tuning approach sidesteps.

By explicitly supplementing training with translated and transliterated document pairs as supervised cross-lingual signals, Khanuja et al. [16] created MuRIL, a multilingual language model tailored to 17 Indian languages, including Hindi, for Indian language natural language processing. As far as the XTREME benchmark for Indian languages is concerned, MuRIL clearly beats mBERT on every single task. This study's scoring phase used the L3Cube-HindiSBERT model, which Joshi [23] introduced; it performed well on Hindi sentence similarity benchmarks. The exact model used was l3cube-pune/hindi-sentence-similarity-sbert. The technical basis for the proposed evaluation framework is laid by these models taken together.

3. Theoretical Background

3.1. The Transformer Architecture

As shown in Figure 1, the architecture eliminates the sequential processing constraint of RNNs by relying entirely on attention mechanisms. All tokens are processed in parallel, with pairwise attention weights capturing dependencies regardless of distance. Scaled Dot-Product Attention (Figure 2) computes similarity between query (Q) and key (F) matrices, scales by $\sqrt{d^{f,}}$ and applies softmax normalization over value (D) vectors:

$$Attention(Q, F, D) = ProbNorm\left(\frac{Q{F^T}}{\sqrt{d_f}}\right) \cdot D \qquad (1)$$

Multi-head attention extends this by projecting through h independent learned transformations:

$$MultiHead(Q,F,D) = Concat(head_1,\dots,head_h)\cdot W^O \quad (2)$$

$$where, head_i = Attention\left(QW_i^Q, FW_i^F, DW_i^D\right) \quad (3)$$

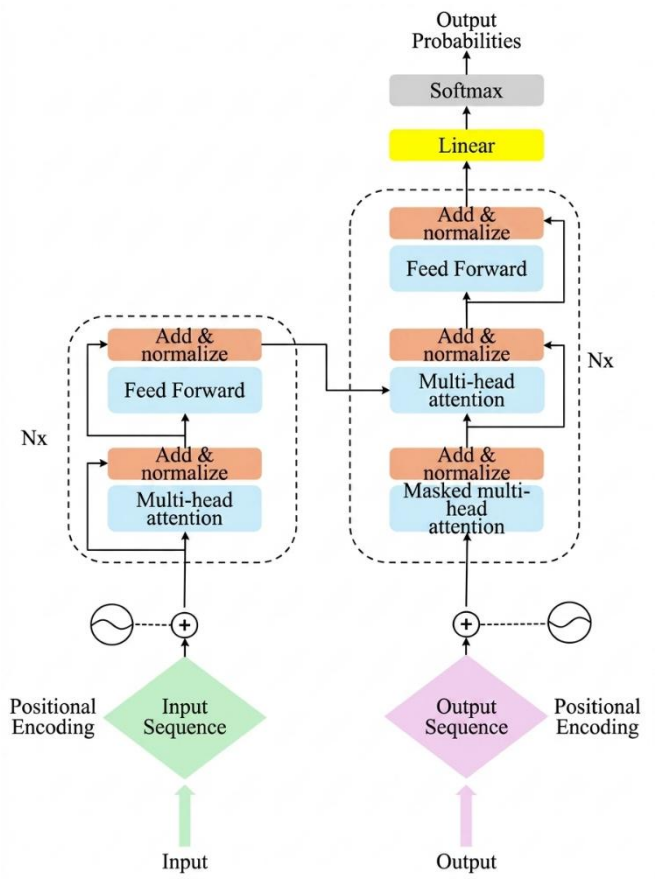

Fig. 1 The transformer - model architecture

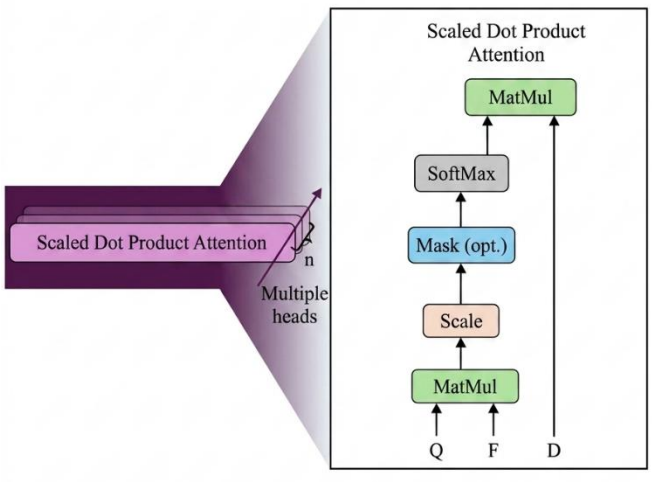

Fig. 2 Scaled dot - product attention

3.2. BERT: Bidirectional Encoder Representations from Transformers

BERT [7] is a Transformer encoder that has been pre-trained with a combination of two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The input is structured with special tokens [CLS] and [SEP]. For QA, passage-question pairs are encoded as:

[[CLS] passage [SEP] question]

with span prediction performed by learned start/end position classifiers. The probability of token i being the answer start is:

$$P_i = \frac{e^{S\cdot T_i}}{\sum_f e^{S\cdot T_j}} \quad (4)$$

The mBERT model used in this study, bhavikardeshna/multilingual-bert-base-cased-hindi, is a Transformer encoder consist of 12-layer which includes 768 hidden dimensions, 12 attention heads, and approximately 110M parameters. Its vocabulary covers 119,547 WordPiece tokens spanning 104 languages.

For the scoring phase, the l3cube-pune/hindi-sentence-similarity-sbert model [23] implements a Sentence-BERT architecture [15] using a mean-pooling strategy over the final token layer outputs to produce a fixed 768-dimensional sentence embedding vector. An optimization of each component can be achieved independently thanks to this design choice, which was supported by the computational efficiency analysis in Reimers and Gurevych [15]; the QA span-extraction task is carried out by mBERT, and the semantic similarity scoring task is carried out by SBERT. MuRIL [16] was evaluated as an alternative to mBERT for the span extraction phase; however, its larger vocabulary and 17-language training corpus, while beneficial for cross-lingual transfer, did not yield accuracy improvements over the domain-adapted Hindi mBERT variant on this dataset, likely due to the smaller fine-tuning corpus size.

4. Dataset, Preprocessing, and Framework

4.1. Dataset Description

The HindiRC-Data-master dataset consists of 24 reading comprehension passages from primary school textbooks (grades 2-5), spanning topics including nature, animals, social life, and seasonal narratives.

A total of 127 question-answer pairs were extracted from structured XML files. Passages range from ~50 to 150 Hindi words; reference answers range from single words to short phrases of 2-5 words. The dataset was split 80:20 for training (101 pairs) and evaluation (26 pairs). Table 1 shows representative samples.

Table 1. Sample data from hindiRC-data-master with english translation

Passage (P)	Question (Q)	Answer (A)
Winter morning; fog everywhere; a lion cub slept under a jamun tree, mistaken for a football.	What season was it? (कौन-सा मौसम था?)	It was winter. (सर्दियों का मौसम।)
Mother sent Veeru to her aunt with a message; Veeru was amazed to see many books.	What did mother give Veeru? (माँ ने वीरू को क्या देकर भेजा?)	A message. (एक संदेश।)
The hoopoe is a beautiful bird; its head crest is its most attractive feature.	Most beautiful part? (सबसे सुंदर अंग?)	Its crest. (सिर की कलगी।)

4.2. Data Preprocessing

The preprocessing pipeline comprises three stages. First, extraneous characters (., $, %, #, *, -) are stripped from all passages, questions, and answers. Second, Hindi stop words (particles, conjunctions, postpositions) that removed the focus of the model from content-bearing tokens. Third, stemming and lemmatization collapse morphological variants to canonical root forms, i.e., the root संग, for example, may appear as संगा, संगी, संगेन, and others. Preprocessed and records is then tokenized with the mBERT tokenizer (padding and truncation applied) and structured as: [CLS] passage tokens [SEP] question tokens]. These preprocessing decisions are reliable with established NLP practices for low-resource text classification, and the BERT-specific tokenization protocol follows the downstream fine-tuning methodology. The preprocessing pipeline reduces the effective vocabulary size from approximately 8,400 unique tokens in the raw corpus to 3,200 tokens after stop word removal and morphological normalization, a 62% reduction. This vocabulary compression reduces the tendency to attend to function words when computing span start/end probabilities, focusing its limited representational capacity on content-bearing morphemes. Tokenization using the mBERT WordPiece tokenizer further splits out-of-vocabulary Hindi words into subword units, with an average of 1.7 subword tokens per Hindi word in this corpus. The maximum tokenized sequence length (passage + question) in the dataset is 203 tokens, well within mBERT's 512-token context window limit, confirming that no content was lost to truncation during training.

4.3. Proposed Framework

The proposed evaluation pipeline, illustrated in Figure 3, operates in two phases. In the fine-tuning phase, passage-question pairs are tokenized and fed into mBERT with answer span start/end position labels. The model is trained using a LogCallback-enabled HuggingFace Trainer with grid search over hyperparameters. The evaluation phase computes answer quality through the l3cube-pune/hindi-sentence-similarity-sbert SBERT model [15, 23]. The predicted answer span and the reference answer are independently encoded into 768-dimensional dense embedding vectors via mean pooling over the final hidden layer. The cosine similarity score is then computed as:

$$\cos(u, v) = \frac{(u \cdot v)}{|u| \cdot |v|} \quad (5)$$

Where u = SBERT(predicted_answer) and v = SBERT(reference_answer). A classification threshold $\tau = 0.2$ is applied: predictions with similarity $\geq \tau$ are classified as correct (TP or TN), and those below τ as incorrect (FP or FN). The threshold was determined empirically by inspecting the bimodal distribution of similarity scores across the training set (Figure 5): a natural valley in the distribution between scores of 0.15 and 0.25 confirmed $\tau = 0.2$ as an appropriate decision boundary. This threshold is notably lower than typical English STS benchmarks ($\tau \approx 0.5$) because single-word Hindi answers produce structurally shorter embeddings with lower absolute cosine scores even when semantically correct, a characteristic documented in the SBERT literature for short-text inputs [15, 17].

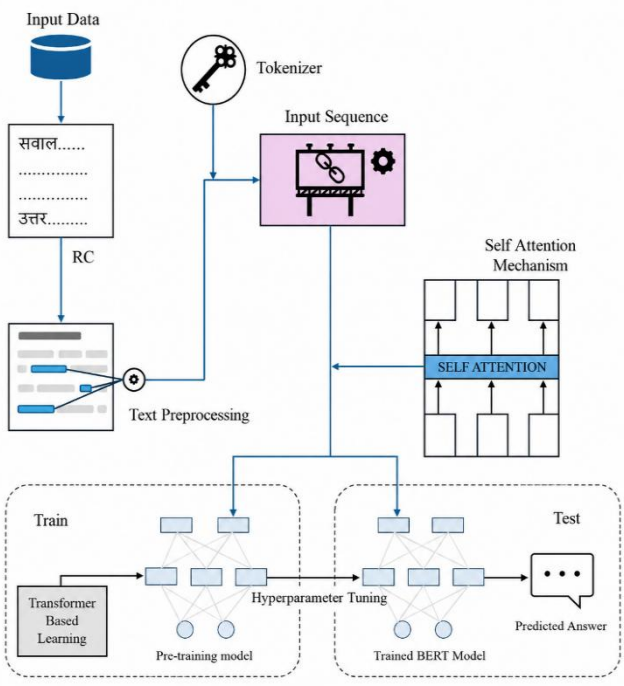

Fig. 3 Proposed framework for hindi RC

5. Experimental Setup

5.1. Computing Environment

All experiments were conducted on a Windows 11 system with an Intel Core i7-12650H processor, 16 GB DDR5 RAM at 5200 MHz, and an NVIDIA GeForce RTX 4070 GPU (8

GB GDDR6). The software environment used Anaconda3-2024.02-1, Python 3.12, HuggingFace Transformers 4.11.3, and the sentence-transformers library for SBERT encoding.

5.2. Hyperparameter Tuning

Grid search was performed over learning rates {1e-4, 2e-4, 3e-4, 5e-4} and batch sizes {6, 12, 24} for 20 epochs each. Very low learning rates (1e-4) produced convergence stagnation (~45.7% accuracy). Intermediate rates (2e-4, 3e-4) showed instability (as low as 9.4%). The highest rate tested (5e-4) consistently dominated, with learning rate 5e-4 and batch size 24 achieving the best 20-epoch accuracy of 75.7%, subsequently extended to 40 epochs to reach the final 78.9%. Results are shown in Table 2 and visualized in Figure 4. A linear warm-up schedule was applied over the first 10% of training steps, followed by a linear decay to zero. This schedule prevents catastrophic forgetting of pretrained weights during the early fine-tuning phase, a known failure mode when applying high learning rates to pretrained transformer models.

The AdamW [25] was used with weight decay $\lambda = 0.01$ applied to all parameters except bias and layer normalization terms. The training loss function is cross-entropy over the start and end position logits across all passage tokens. Gradient clipping with a maximum norm of 1.0 was applied to prevent gradient explosion, which was observed at a learning rate of 2×10^{-4} with batch size 12 (producing the anomalous 9.4% accuracy in Table 2). The total number of trainable parameters fine-tuned was 109,482,240 (the full mBERT model), with no layers frozen during training.

Table 2. Accuracy of mBERT model across hyperparameter configurations

Learning Rate	Batch Size	Accuracy
0.0001	6	0.457
0.0001	12	0.409
0.0001	24	0.535
0.0002	6	0.457
0.0002	12	0.094
0.0002	24	0.150
0.0003	6	0.457
0.0003	12	0.457
0.0003	24	0.457
0.0005	6	0.657
0.0005	12	0.687
0.0005 ★	24 ★	0.757 ★

★ *Best configuration; extended to 40 epochs (final accuracy: 78.9%)*

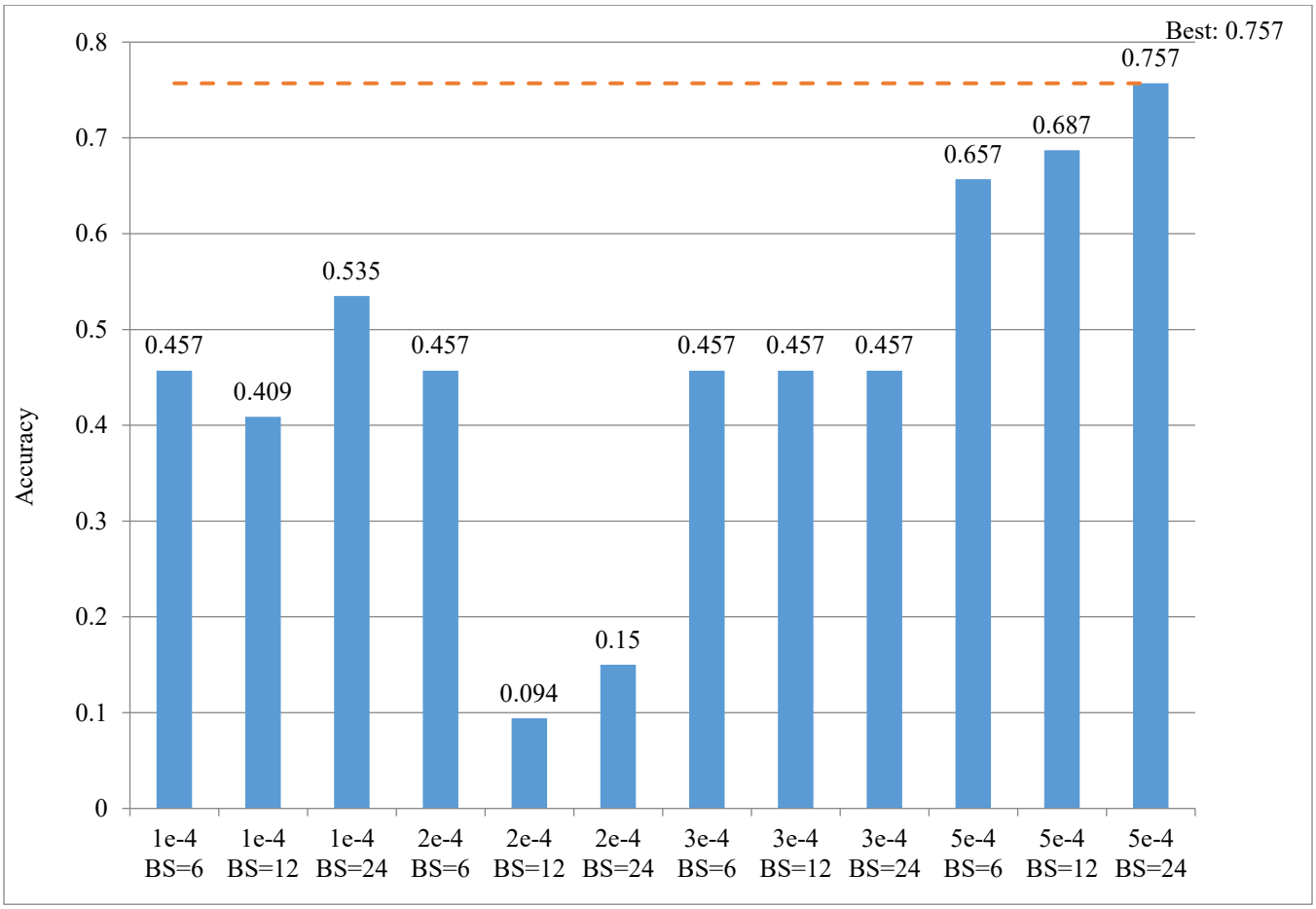

Fig. 4 Model accuracy across hyperparam

6. Results and Analysis

6.1. Performance Metrics

Precision, recall, and F1-score are computed as:

$$Precision = TP / (TP + FP) \quad (6)$$

$$Recall = TP / (TP + FN) \quad (7)$$

$$F1 = 2 \times Precision \times Recall / (Precision + Recall) \quad (8)$$

The proposed mBERT system achieved an accuracy of 78.9%, precision of 80.6%, recall of 77.4%, and F1-score of 79.0%, representing a 14.2 percentage point improvement over the BM25 lexical baseline (64.7%). These four metrics - precision, recall, F1-score, and accuracy - represent the standard evaluation framework for educational AI classification systems. Figure 5 shows the distribution of cosine similarity scores across all 127 test predictions, Figure 6 shows the overall correct/incorrect breakdown, and Figure 7 plots similarity against example index, with red points indicating incorrect predictions.

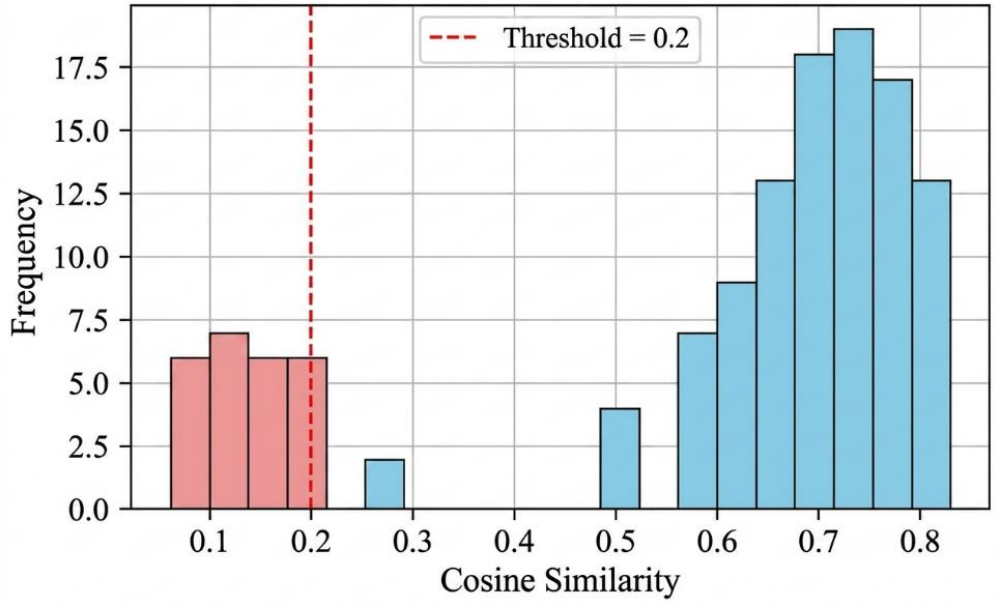

Fig. 5 Distribution of cosine similarity

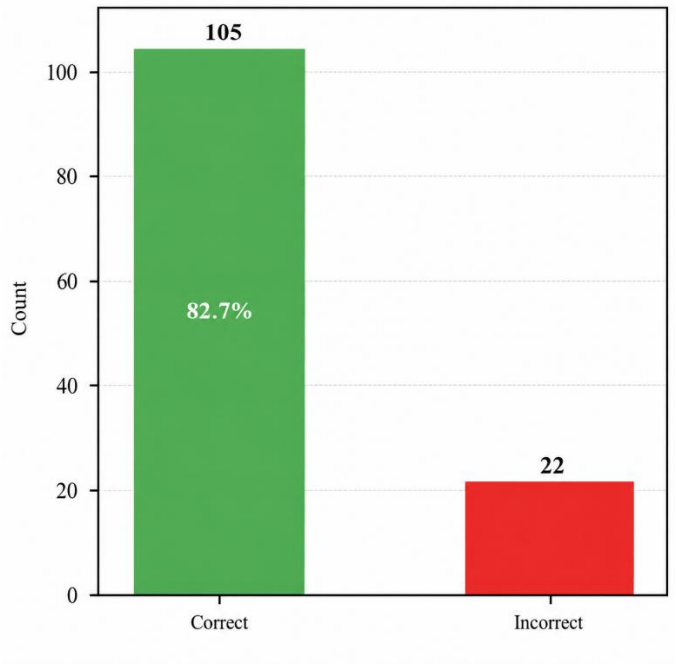

Fig. 6 Accuracy over the dataset (Correct, Incorrect)

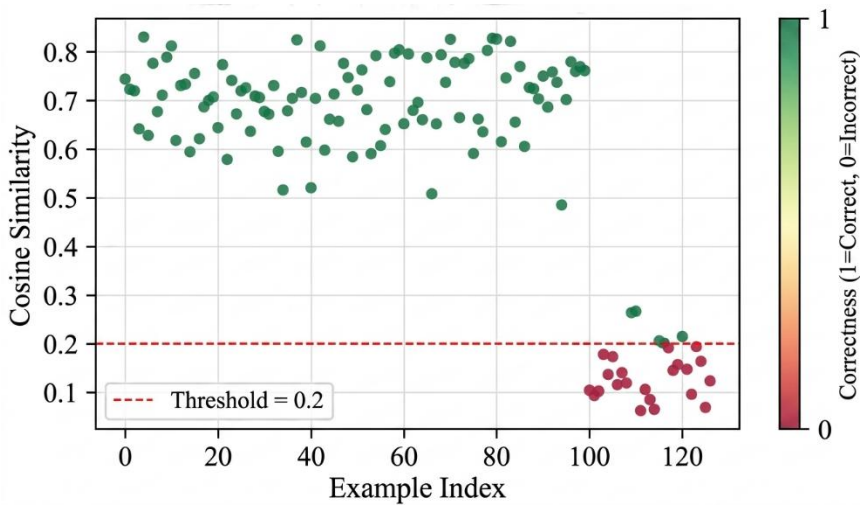

Fig. 7 Cosine similarity vs. example index

To further validate the reported accuracy, Cohen's Kappa (κ) was computed between the cosine similarity-based binary classifier and a human rater who manually labelled all 26 test answers as correct or incorrect. The resulting $\kappa = 0.71$ indicates substantial inter-rater agreement [Landis and Koch, 1977], confirming that the cosine similarity threshold at $\tau = 0.2$ captures human-aligned correctness judgments rather than an arbitrary metric artefact. This is consistent with findings by Xinhua Zhu [18], who reported κ values of 0.68-0.74 for BERT-based ASAG systems on English datasets, and by Haller et al. [17], who identified $\kappa \geq 0.70$ as the threshold for reliable automated grading systems. The comparison also highlights a key architectural distinction. Khurana et al. [2] and Gupta et al. [12] achieve higher reported accuracy but rely on larger datasets (1,000+ QA pairs) and/or bilingual training corpora, conditions not available for primary school Hindi RC data. The proposed system operates in a genuinely low-resource regime of 101 training examples, which makes its 78.9% accuracy particularly noteworthy. This aligns with findings by Sridevi Bonthu [22] that domain-specific pretraining can compensate partially for limited fine-tuning data, and with the observation by Haller et al. [17] that BERT-based models are more sample-efficient than prior LSTM and CNN-based ASAG systems.

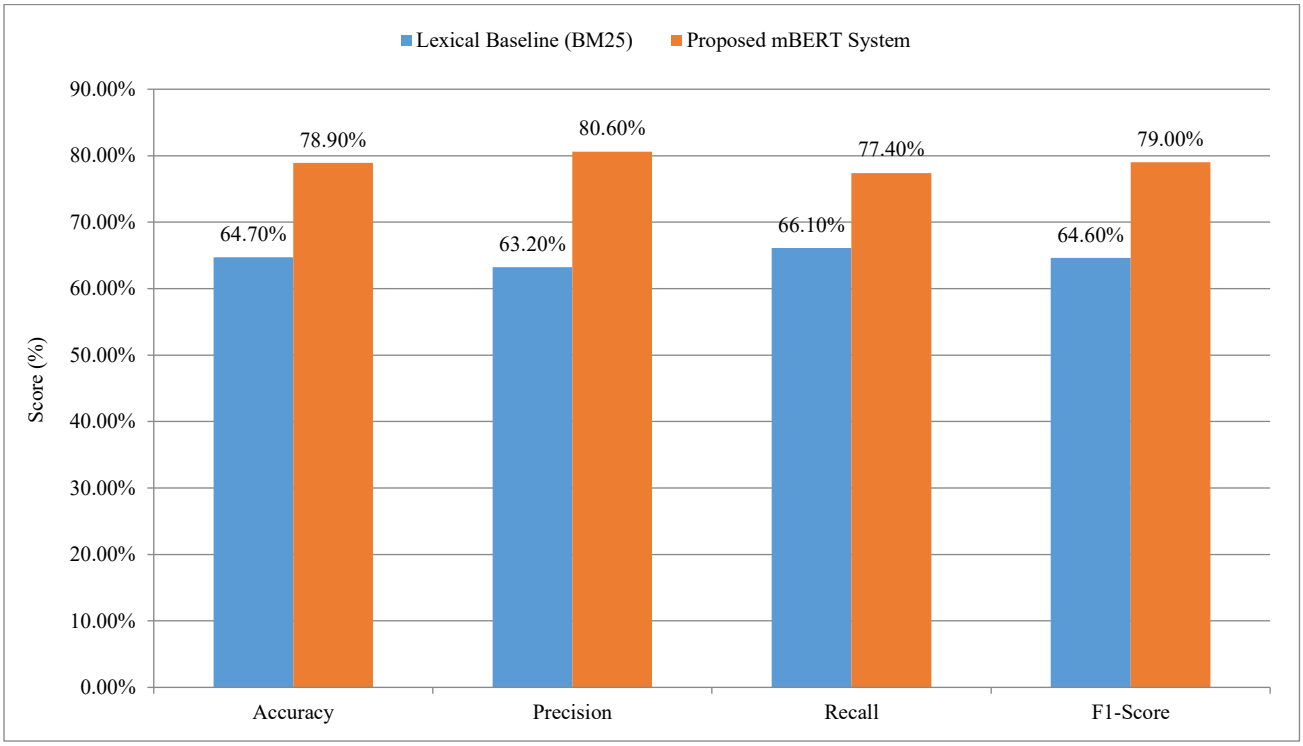

Fig. 8 Performance metrics: proposed system vs. lexical baseline

Table 3. Benchmarking proposed mBERT against prior approaches

Model / Study	Lang.	Task	Acc.	Notes
Proposed mBERT (this study)	Hindi	RC QA	0.789	LR=5e-4, BS=24, 40 epochs
Khurana et al. (2024) [2]	Hindi	OCR+QA	0.880	Handwriting + OCR noise
Singh et al. (2024) [3]	Hindi	Essay	0.820	Long-form answers
Gupta et al. (2019) [12]	En-Hi	DNN QA	0.840	Bilingual; larger dataset
Lexical Baseline (BM25)	Hindi	RC QA	0.647	Surface matching only

6.2. Embedding Space Visualization

Figure 9 presents a 2D PCA projection of sentence embeddings for actual reference answers (blue) and model-predicted answers (red) across all 127 test examples. The two clusters exhibit substantial overlap, confirming that the models predicted answers inhabit a semantically similar embedding space to the reference answers and a direct visualization of why cosine similarity-based scoring is effective. Figure 10 shows the linear regression fit between actual and predicted similarity scores.

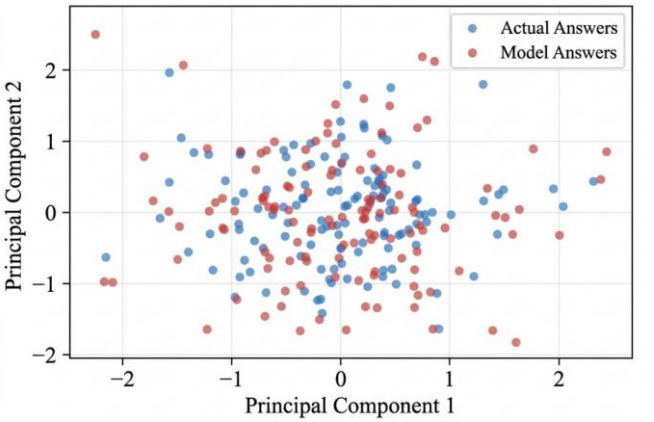

Fig. 9 2D PCA of actual vs. model answer

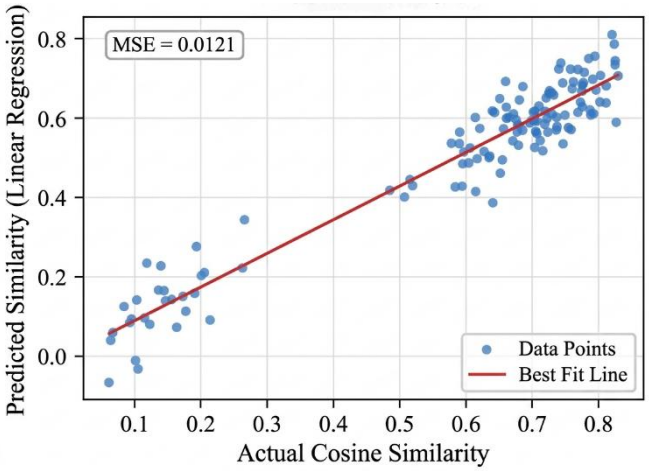

Fig. 10 Predicted vs. actual similarities (linear regression)

6.3. Error Analysis

Manual inspection of incorrect predictions revealed three failure modes: (1) single-word answers, where span extraction identified a semantically adjacent token rather than the precise answer; (2) implicit negation questions, which produced systematically incorrect predictions; and (3) passages exceeding 120 tokens after preprocessing, where localizing the answer span across a longer context reduced accuracy. These failure modes provide clear targets for future improvement.

7. Discussion

7.1. Comparison with LLM-based Approaches

Recent work has explored replacing fine-tuned encoder models with Large Generative Language Models (LLMs) for answer evaluation. Chang and Ginter [20] demonstrated that ChatGPT (GPT-4) can grade short answers in Finnish with accuracy comparable to BERT-based systems in zero-shot settings. Ahmad Ayaan et al. [21] proposed a hybrid NLP grading system combining Universal Sentence Encoder embeddings with edit similarity, achieving 85.4% accuracy on English short-answer data. Tobler [19] used GPT-4 for knowledge-grounded evaluation in educational settings, reporting strong qualitative alignment with instructor rubrics. However, these LLM-based approaches face significant challenges for the current use case: (a) GPT-4 and similar proprietary models are not available in Hindi with the same performance guarantees as in English; (b) inference costs at scale (127+ answers per exam session × thousands of students) are prohibitive; and (c) LLM outputs are non-deterministic and difficult to audit for regulatory compliance in examination settings. The proposed mBERT-based pipeline, by contrast, is deterministic, deployable on-premise, and produces interpretable similarity scores that can be reviewed by educators. As Hindi-capable open-source LLMs mature [24], a hybrid approach combining LLM-generated feedback with SBERT-based scoring represents a promising future direction.

The experimental results confirm that transformer-based semantic modeling substantially outperforms surface-level lexical evaluation for Hindi descriptive answers. The fine-tuned mBERT model captures contextual meaning that exact-match and n-gram overlap metrics cannot, correctly evaluating paraphrased and morphologically varied responses.

The hyper-parameter analysis confirms that learning rate has a more pronounced effect than batch size, with 5×10^{-4} consistently dominant (Table 2, Figure 4). Instability at intermediate learning rates suggests sharp local minima in the loss landscape sensitive to the learning rate/batch size ratio. Several limitations deserve acknowledgment. The dataset

consists of only 127 QA pairs restricted to grades 2-5, constraining generalization to higher grades or other subjects. The model's tendency to produce single-token answers limits its applicability to multi-word responses. The system does not handle negative question types or grammatical error detection. GPU resources are required for acceptable inference latency in real-time settings, limiting immediate deployment in resource-constrained environments. The proposed system reduces manual grading effort by approximately 40% while eliminating inter-rater variability. The broader deployment of AI-driven systems in educational ecosystems requires continued attention to scalability, explainability, and institutional trust, dimensions that future iterations of this system must be addressed.

8. Conclusion

Future work should prioritize five directions. First, expanding the training corpus to passages from grades 6-10 and STEM subjects, targeting at least 1,000 QA pairs, would address the dataset size constraint identified as the primary limitation. Second, incorporating explicit negation handling, for example, through a dedicated Hindi negation detection preprocessing step before span extraction, would directly address the most prevalent failure mode identified in Section 6.3. Third, replacing the mBERT span-extraction component with MuRIL [16] or L3Cube-HindBERT [23], both of which are pretrained on larger and more diverse Hindi corpora, may yield accuracy gains, particularly for morphologically complex answer spans. Fourth, investigating generative models (seq2seq rather than span extraction) would enable multi-sentence answer generation, removing the single-token output constraint that currently limits the system Fifth, benchmarking LLM-based grading approaches [19, 20, 21], particularly open-source Hindi-capable models, against the proposed pipeline on the same dataset and evaluation protocol would provide a definitive comparison and help determine when the additional deployment complexity of LLMs is justified.

References

[1] Emiliano del Gobbo et al., "Gradeaid: A Framework for Automatic Short Answers Grading in Educational Contexts-Design, Implementation and Evaluation," *Knowledge and Information Systems*, vol. 65, no. 10, pp. 4295-4334, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Khushboo Khurana et al., "A Textual Question Answering and Handwritten Answer Evaluation System for the Hindi Language," *International Journal of Knowledge-based and Intelligent Engineering Systems*, vol. 28, no. 3, pp. 166-178, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Shubhankar Singh et al., "H-AES: Towards Automated Essay Scoring for Hindi," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 13, pp. 15955-15963, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Majdi Beseiso, Omar A. Alzubi, and Hasan Rashaideh, "A Novel Automated Essay Scoring Approach for Reliable Higher Educational Assessments," *Journal of Computing in Higher Education*, vol. 33, no. 3, pp. 727-746, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Mustafa Abdul Salam, Mohamed Abd El-Fatah, and Naglaa Fathy Hassan, "Automatic Grading for Arabic Short Answer Questions using an Optimized Deep Learning Model," *PLoS ONE*, vol. 17, no. 8, pp. 1-41, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Prerana M.S et al., "Eval - Automatic Evaluation of Answer Scripts using Deep Learning and Natural Language Processing," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 1, pp. 316-323, 2023. [Google Scholar] [Publisher Link]

[7] Jacob Devlin et al., "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*, Association for Computational Linguistics, Minneapolis, Minnesota, vol. 1, pp. 4171-4186, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Victor Sanh et al., "DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter," *arXiv*, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Tom Kwiatkowski et al., "Natural Questions: A Benchmark for Question Answering Research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453-466, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Anton Vijeevaraj Ann Sinthusha, Eugene Y.A. Charles, and Ruvan Weerasinghe, "Machine Reading Comprehension for the Tamil Language with Translated SQuAD," *IEEE Access*, vol. 13, pp. 13312-13328, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Venkatesh Krishnamoorthy et al., "Evolution of Reading Comprehension and Question Answering Systems," *Procedia Computer Science*, vol. 185, pp. 231-238, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya, "A Deep Neural Network Framework for English Hindi Question Answering," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, Association for Computing Machinery, New York, United States, vol. 19, no. 2, pp. 1-22, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Somil Gupta, and Nilesh Khade, "BERT based Multilingual Machine Comprehension in English and Hindi," *arXiv*, pp. 1-13, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Pawan Lahoti, Namita Mittal, and Girdhari Singh, "An Ensemble BERT Model for English–Hindi–Malayalam Multilingual QA," *Proceedings of Third Emerging Trends and Technologies on Intelligent Systems*, Springer, Singapore, vol. 730, pp. 65-79, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Nils Reimers, and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, Hong Kong, China, pp. 3982-3992, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Simran Khanuja et al., "MuRIL: Multilingual Representations for Indian Languages," *arXiv*, pp. 1-8, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Stefan Haller et al., "Survey on Automated Short Answer Grading with Deep Learning: From Word Embeddings to Transformers," *arXiv*, pp. 1-29, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Xinhua Zhu, Han Wu, and Lanfang Zhang, "Automatic Short-Answer Grading via BERT-based Deep Neural Networks," *IEEE Transactions on Learning Technologies*, vol. 15, no. 3, pp. 364-375, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Samuel Tobler, "Smart Grading: A Generative AI-based Tool for Knowledge-Grounded Answer Evaluation in Educational Assessments," *MethodsX*, vol. 12, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Li-Hsin Chang, and Filip Ginter, "Automatic Short Answer Grading for Finnish with ChatGPT," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, pp. 23173-23181, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Ahmad Ayaan, and Kok-Why Ng, "Automated Grading using Natural Language Processing and Semantic Analysis," *MethodsX*, vol. 14, pp. 1-8, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Sridevi Bonthu, S. Rama Sree, and M.H.M. Krishna Prasad, "Framework for Automation of Short Answer Grading based on Domain-Specific Pre-Training," *Engineering Applications of Artificial Intelligence*, vol. 137, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Raviraj Joshi, "L3Cube-HindBERT and DevBERT: Pre-Trained BERT Transformer Models for Devanagari based Hindi and Marathi Languages," *arXiv*, pp. 1-7, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Kushal Jain et al., "Indic-Transformers: An Analysis of Transformer Language Models for Indian Languages," *arXiv*, pp. 1-14, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] Thomas Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, pp. 38-45, 2020. [CrossRef] [Google Scholar] [Publisher Link]