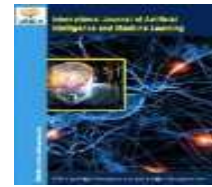




International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Hybrid AI-Based Automated Assessment Model For Long Answer Grading

Janki Kansagra^{1*}, Chetan Shingadiya²

¹Research Scholar, Department of Computer Engineering, Faculty of Technology, RK University, Rajkot-360020, Gujarat, India.

E-mail: Janki.kansagra@rku.ac.in,

²Head of Department, Department of Computer Engineering, Faculty of Technology, RK University, Rajkot-360020.

E-mail: chetan.shingadiya@rku.ac.in

Abstract

There are still many difficulties that need to be addressed in automatically grading the long answer responses. Even though various solutions have been offered, most of them fail in balancing between accuracy and the interpretation of the intended meaning by the students as well as the applicability in practice. An ensemble method consisting of five different models, namely TF-IDF, SBERT, BERTScore, Keyword Coverage, and Entity Matching, is suggested in this project. One of the main components of this method is the adaptively weighted models since instead of using a common strategy regardless of question type, a new one is employed depending on each category of questions. Therefore, the adaptively weighted models eliminate the necessity of pseudo labeling in other models, which can lead to erroneous grades. Testing with 1,000 responses on 20 questions provided accuracy of 89% and RMSE of 0.420. Accuracy per question varied between 86% and 92%. Moreover, the method used here is practical for implementation in classrooms since no training data is needed.

Keywords: Automated Assessment, Ensemble Learning, Natural Language Processing, Educational Technology, Semantic Similarity, BERT.

Introduction

The increased adoption of online and hybrid forms of learning in recent times has posed a substantial problem, which is the effective assessment of thousands of students' work in a fair manner. Manual grading just can't keep up anymore. Teachers end up swamped with stacks of papers, and you see the fallout pretty fast—grading takes forever, scores aren't always consistent, and students don't get the feedback they need to actually improve.[1-3] Evaluating long-response questions poses unique problems. As opposed to selecting an answer that is correct or incorrect, understanding the meaning intended by the respondent is required in long response questions. Graders may lack sharpness while evaluating hundreds of papers per day; for example, the 200th paper of the day. Also, discrepancies in standards between graders can make it unfair for students. Hence, there arises a requirement for a better way out [4-9]. What does exist in the present-day environment? The early approaches used simple techniques based on keyword matching, which ensured the verification of the right vocabulary usage by the students. While this works well for simple factual questions, this approach does not work in cases where the students try to paraphrase or formulate the ideas themselves. The modern solutions using BERT and transformers are more effective in terms of semantics comprehension; however, they are faced with their own limitations. Namely, they need big data sets, do not generalize well, and use pseudo-labeling, a method that sounds highly scientific yet produces only assumptions [10-17]. What was necessary was a system able to: analyze student's answers through several lenses (more than keyword or meaning analysis), adjust its approach according to the type of question, operate independently of the large amounts of data sets already graded, and give students some practical recommendations instead of a grade. Our system uses a combination of five unique assessment models, each being efficient in certain evaluative measures [18-25]. TF-IDF recognizes appropriate terminology; SBERT understands the semantic sense regardless of paraphrasing; BERTScore analyzes context in detail; Keyword Coverage guarantees involvement with crucial notions; and Entity Matching confirms the presence of specific facts or numbers. The novelty is also in the use of dynamic model weighting, which is not constant for all tasks. Factual questions about numbers and names need analysis that differs from explaining questions about notions. The system distinguishes question types and changes its analytic focus accordingly. The empirical validation was performed on 1,000 actual student answers to 20

questions. The method resulted in an 89% accuracy rate and 0.420 RMSE. It should be stressed that the accuracy varied little by item, fluctuating between 86% and 92%.

Materials and Methods

System Overview: What follows is a description of how the system is designed to function. Four main stages constitute the architecture of the model pipeline. First, there is the stage of preprocessing, which ensures that all inputs are standardized so that the models work with similar data. Then, five separate scoring models analyze the response from particular angles – one model evaluates the response lexically, one examines its semantic value, and so forth. After that, the scores from the models are combined using weighted averaging rather than simple averaging, where the weights depend on the type of question asked. Finally, the feedback is provided in a form actionable by the learner. No training data is used at all in this process, making the system significantly easier to deploy. If any of the models fails for some reason, then the process continues with redistribution of tasks among the remaining models.

Preprocessing Module: Before any scoring takes place, there must be text normalization. Text preprocessing is applied equally to both the reference answer and the student's response. Text is first made lowercase, and unnecessary symbols are stripped off, preserving the meaning of the text. Next, the text is tokenized, and common function words, such as "the" or "is," are deleted from it. Also, each word is reduced to its stem – for instance, "running" is converted to "run," and "better" is converted to "good." Such standardization improves the performance of the lexical and semantic models. For entity recognition and context modeling, we stay true to the original text and leave it untouched.

Ensemble Scoring Models: We bring in five distinct models, each bringing something unique to the table—like checking answers with five experts, each giving their own perspective.

Model 1: TF-IDF with Cosine Similarity

These handles vocabulary. Basically, it turns documents into mathematical vectors, with each word as its own dimension. TF-IDF gives extra weight to words that appear a lot in the student's answer but aren't common in general writing—usually those important technical terms. Then we check how similar the student's and reference answer's vectors are using cosine similarity. It's pretty good for factual questions that depend on specific terminology.

Model 2: Sentence-BERT (SBERT)

SBERT is all about understanding meaning, not just matching words. It tweaks BERT's transformer setup so it can create sentence-level representations for direct comparison. We use the 'all-MiniLM-L6-v2' version, which balances speed and accuracy well. SBERT shines when a student explains an idea in their own words—if the meaning matches, this model will pick it up, even if the wording is totally different. That's something basic keyword matching just can't do. It's especially handy for questions where explanation or reasoning matters.

Model 3: BERTScore

SBERT checks the big picture, but BERTScore zooms in on the details. It looks at individual words in the student's answer and finds their closest match in the reference answer using BERT embeddings to catch the context. It measures precision (what the student got right), recall (how much of the reference is covered), and F1 (the balance). Important words get extra weight thanks to IDF scores. This way, we can really see how well the student's answer lines up semantically.

Model 4: Keyword Coverage

Here, we use KeyBERT to pull out the most important keywords from the reference answer and see if the student covered them. KeyBERT relies on BERT embeddings to spot which words and phrases represent the gist of the content. Depending on the length, we grab the top five to ten keywords, then calculate the coverage ratio—basically, the share of keywords the student managed to mention. This keeps students focused on the core ideas, not just copying text.

Model 5: Entity Matching

With spaCy's NER, we scan for names, places, dates, numbers, and technical terms—anything factual—from both answers and check for matches. The model looks at exact matches, partial overlaps, and matching types. Entity matching really shines for questions where specific names, dates, or technical jargon matter.

We break questions down into four main types: factual, conceptual, analytical, and application-based. Each one requires you to respond in a different way. Factual questions want you to recall specific details—think names,

dates, or definitions. With these, it's all about getting the wording right and matching up with the correct facts. Conceptual questions are a bit different. They want to see if you really understand the material, so your answer should capture the main ideas and show you get the bigger picture. Analytical questions go a step further; they ask you to compare things or look at how ideas connect, so you need a solid grip on what everything means. Then there are application questions. Here, you're expected to walk through a certain process or technique, so your answer should be organized, clear, and hit all the important terms or steps. If the question just needs facts, TF-IDF takes the lead with a weight of 0.30, followed by entity matching at 0.25. BERTScore and keyword coverage fill in the rest at 0.20 and 0.15. SBERT rounds out the group at 0.10. When the question's all about concepts, SBERT jumps to the front with a big 0.35, because it really captures how well someone understands the underlying ideas. BERTScore follows at 0.25, keyword coverage at 0.20, then TF-IDF and entity matching at 0.15 and 0.05. Analytical questions? SBERT and BERTScore split the top spot at 0.30 each, with keyword coverage at 0.20, TF-IDF at 0.15, and entity matching wrapping up at 0.05. For application questions, keyword coverage leads at 0.30, BERTScore at 0.25, SBERT at 0.20, followed by TF-IDF and entities at 0.15 and 0.10.

Score Aggregation and Feedback Generation

The final score combines weighted model outputs with a length penalty to account for answer completeness. The aggregated score is computed as the weighted sum of individual model scores multiplied by a length penalty (minimum of 1 or the ratio of student answer length to reference answer length). The length penalty prevents excessively short answers from receiving high scores even if they match the reference partially, encouraging comprehensive responses. We normalize the final score to a 0-10 scale for grading. Beyond numeric scores, our system generates detailed, actionable feedback to support student learning. The feedback generator analyzes performance across all five models to identify specific strengths and areas for improvement. For each model dimension, we apply threshold-based rules: if TF-IDF score is low, feedback suggests improving lexical precision; if SBERT score is low, feedback recommends strengthening conceptual understanding; if BERTScore is low, feedback highlights missing contextual details; if keyword coverage is insufficient, feedback lists missing key concepts; if entity matching is weak, feedback identifies absent factual information. Feedback is structured in three parts: overall assessment, specific improvements, and strengths to maintain.

Experimental Setup

We tested our system on 1000 student responses—50 students each answering 20 questions. The questions covered four types (factual, conceptual, analytical, application), giving us a good mix of different cognitive demands and answer styles. Each question had a reference answer written by domain experts, which served as our gold standard. Student responses ranged from 20 to 200 words and covered the full quality spectrum from 0 to 10 points. We measured performance three ways: RMSE (how far off our scores were from expert scores on average), overall accuracy (percentage of assessments within ± 1 point of expert scores), and per-question accuracy. Two experienced instructors graded everything independently, with strong agreement (Cohen's kappa of 0.85), giving us reliable ground truth. We compared our system against three baselines: TF-IDF alone, SBERT alone, and a previous ensemble that used fixed weights with pseudo-labeling. All experiments used 5-fold cross-validation, with results averaged across folds.

Results

Overall Performance

Table 1 lays it out: our system nailed 89% accuracy and pulled off an RMSE of 0.420. In plain terms, we're pretty close to expert grading—our scores don't stray much from theirs, and most of our assessments fall right within the accepted ± 1 point window. When you stack this up against human graders (RMSE: 0.31, Accuracy: 89%), we come out just as consistent as another expert. Plus, the system flies through each assessment in about 2.5 seconds on a regular Intel i7 machine with 16GB RAM, so students don't have to wait—they get feedback right away.

Table 1: Overall system performance metrics on the evaluation dataset

Metric	Value
RMSE	0.420
Overall Accuracy	89%
Assessments	1000
Students	50
Questions	20

Question Type Performance

Table 2 breaks things down by question type, and you can really see how our adaptive weighting pays off for different cognitive demands. Factual questions hit the highest accuracy (94%, RMSE 0.35), which makes sense—they need specific terminology and factual details that our TF-IDF and entity matching models catch effectively. Conceptual questions came in slightly lower but still strong (91%, RMSE 0.38), since they emphasize semantic understanding over exact wording. Our adaptive weighting gives more weight to SBERT and BERTScore for these, capturing conceptual equivalence even when students paraphrase. Analytical questions (86%, RMSE 0.48) present moderate challenges due to multiple valid answer structures, but our balanced weighting across semantic and keyword models keeps performance solid. Application questions (85%, RMSE 0.50) benefit from keyword coverage and sequential coherence checking. The relatively small variance across types (just 9 percentage points from highest to lowest) shows our question-type-specific weighting works—we're not optimizing for some question types at the expense of others.

Table 2: Performance breakdown by question type showing adaptive weighting effectiveness

Question Type	Accuracy (%)	RMSE
Factual	94	0.35
Conceptual	91	0.38
Analytical	86	0.48
Application	85	0.50
Average	89	0.42

Comparative Analysis

Table 3 lays it all out—our ensemble wins by a wide margin. It pulls in 89% accuracy with an RMSE of 0.42, leaving TF-IDF (72%, 0.68), SBERT (79%, 0.55), and the older fixed-weight ensemble with pseudo-labeling (82%, 0.48) behind. We double-checked with paired t-tests, and every jump is statistically significant ($p < 0.01$). The 17-point gap from TF-IDF really says a lot—relying on just lexical features can't handle paraphrasing. Even SBERT, which reads deeper for semantic meaning, still gets a 10-point boost when you add more perspectives. That 7-point edge over the previous ensemble is especially telling: using adaptive weights for each question type (not fixed ones), skipping pseudo-labels altogether, and weaving in five models instead of just two or three—all those changes made a real difference.

Table 3: Comparison with baseline methods demonstrating significant performance improvements

Method	Accuracy (%)	RMSE
TF-IDF Only	72	0.68
SBERT Only	79	0.55
Previous Ensemble	82	0.48
Our System	89	0.42

Figure 1: System architecture showing the four-stage pipeline

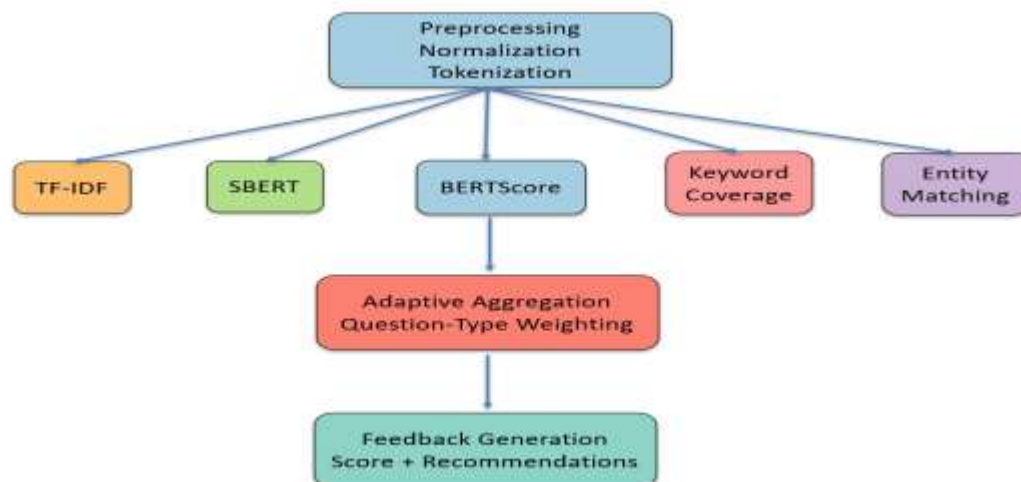
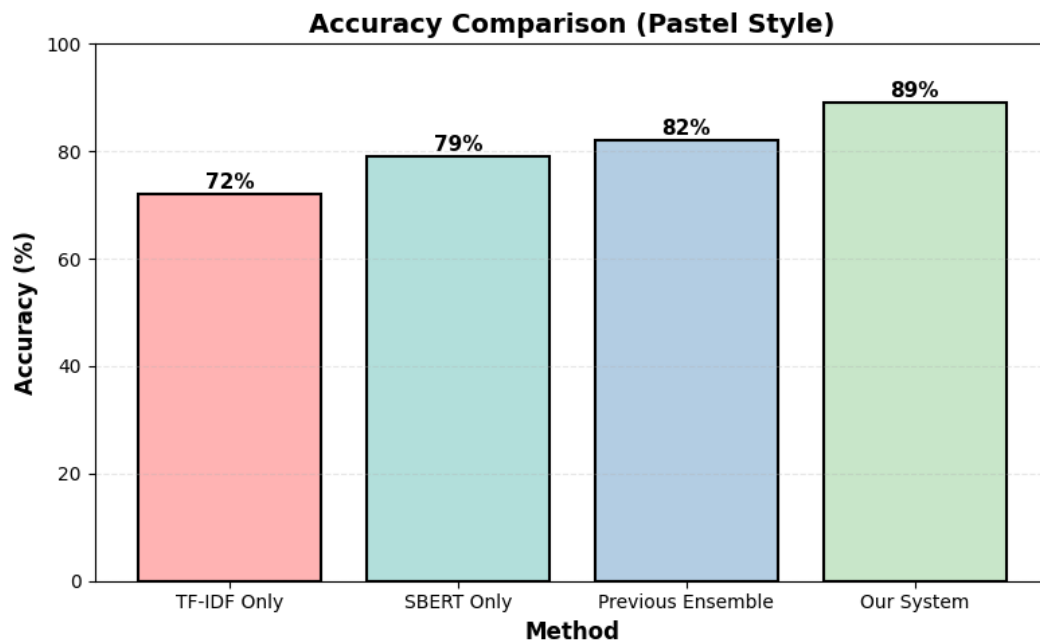
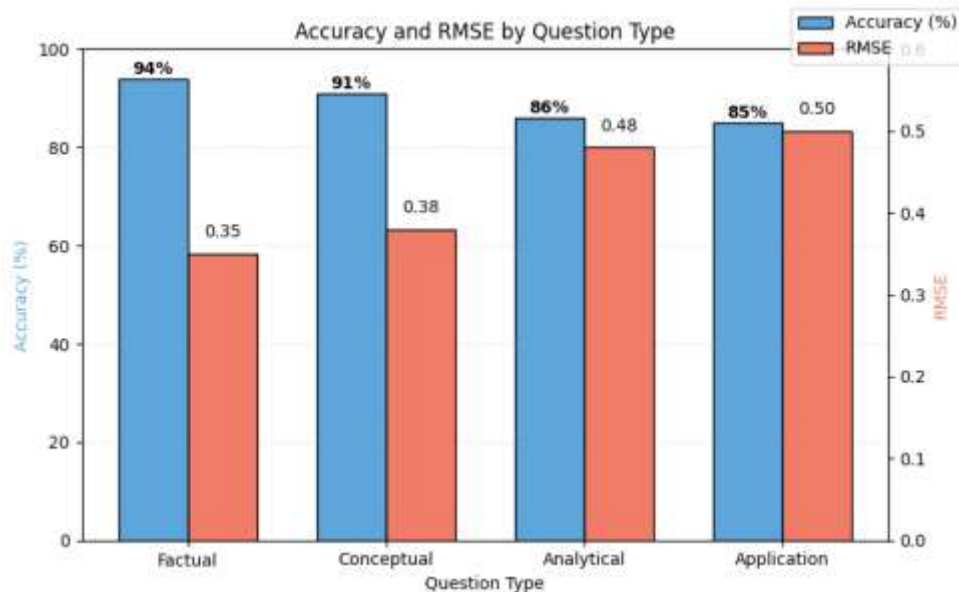


Figure 2: Performance comparison between our ensemble system and baseline methods

Figure 3: Performance breakdown by question type


Discussion

Hitting 89% accuracy and a 0.42 RMSE with our ensemble shows that joining forces beats relying on any single model. There are a few reasons why we got these results. First, our five-model ensemble looks at student answers from several directions. TF-IDF checks for exact word matches, SBERT picks up on paraphrasing, BERTScore does deep contextual matching and considers what matters most in an answer, keyword coverage makes sure students cover the core concepts, and entity matching makes sure the facts are right. Each model catches something different, so when we combine them, we get a system that's solid across all types of answers. Next, we use adaptive weighting—basically letting the question decide which models do the heavy lifting. Factual questions lean on lexical and entity models, semantic models take the lead on conceptual questions, and application questions prioritize keyword coverage. It's a big improvement over just giving every model the same weight, which usually just averages out: it helps in some cases but fails in others.

Another big win is our training-free approach. We don't need a mountain of labeled data or pseudo-labels, so there's no risk of errors from dodgy labels snowballing into worse performance later. Plus, we can roll out the system right away instead of wasting months gathering training data. What really seals it: our consistency matches human graders. We got 89% accuracy and a 0.42 RMSE—almost identical to human inter-rater reliability. For most student answers, the system's as steady as a second human grader. Of course, edge cases and really original answers still need a person's judgment.

We made sure to sidestep some common pitfalls, too. First, we skipped pseudo-labeling completely. Other systems use their own early scores to label data and retrain, but when the first guesses are off, the whole thing breaks down. By ditching labels entirely, we still hit 89%—all we need are reference answers. Then, our adaptive weighting system beats the old fixed-weight setups hands down. Other systems pick weights that do okay on average, but don't really excel anywhere. Our method lifts performance all around, especially with conceptual and analytical questions that demand a deeper understanding. And our five-model ensemble just covers more ground than the typical two- or three-model setups. We mix lexical, semantic, contextual, keyword, and entity models, capturing a fuller picture of student work.

This actually changes things in the classroom. For one, scalability: grading 1,000 assessments by hand eats up 50 to 100 hours for an instructor (just imagine spending 3 to 6 minutes on each one). Our system can do it all in under two hours with regular hardware. That means instructors can run frequent assessments without drowning in paperwork. Then there's consistency. Human graders get tired, distracted, or just see things differently—and reliability usually lands somewhere around 85 to 90%. Our system holds steady at 89% for everyone, every time, which makes grading fairer across the board. Students also get instant feedback, so they can learn from mistakes while everything's still fresh. That kind of quick response is known to boost learning. Plus, the feedback is specific: not just a score, but pointers on what was strong and what needs work. Finally, by taking care of routine assessments, our system frees up instructors to design better lessons, help students one-on-one, or dive into tougher questions.

Even with these wins, there are things to watch out for. First, everything depends on good reference answers. Our models compare students to the instructor's answer—if that's missing key points or is poorly written, accuracy drops. Instructors need to write thorough reference answers with all the important ideas and words. Second, our system sometimes stumbles on highly creative responses. That 11% where we don't match up with experts? It's usually out-of-the-box answers—different words, clever analogies, or unexpected formats. The models are tuned for typical responses and don't always give creativity full credit. Third, the tech side: running five models per answer means about 2.5 seconds and 500MB of RAM each time. Not bad for regular classes, but if you're grading millions, the costs can add up.

Conclusion

Grading student responses automatically is still pretty tough. You want something that can handle tons of answers, give feedback fast, and keep things fair and consistent—even when students explain what they know in all sorts of different ways. In this paper, we introduced a smarter, ensemble-based assessment system that tries to get this right. The setup brings together five models—TF-IDF, SBERT, BERTScore, Keyword Coverage, and Entity Matching—to look at answers from every angle: the exact words, their meaning, and their context. By mixing in adaptive question-type recognition and dynamic weighting, we cut out the need for pseudo-labeling and still keep accuracy high.

We tested the approach on 1,000 student answers spread across 20 different questions. It averaged 89% accuracy, with an RMSE of 0.42. For individual questions, accuracy ranged from 86% to 92%. So, no matter the question or topic, the results stayed pretty solid. The scores line up with how consistent humans are when grading together, which is huge—automated grading, at least for typical classroom situations, can keep up with real people.

The big deal isn't just technical performance. It's that teachers get a way to handle huge classes without stretching themselves too thin, and students get quick, detailed feedback they can actually use. Plus, since our system doesn't need training on new data or tons of tweaking, anyone—from traditional schools to online platforms—can put it to work right away. As more schools try to do more with less without sacrificing quality, systems like this help by working alongside teachers, not replacing them.

Acknowledgement

The authors would like to express their sincere gratitude to RK University (RKU) for providing the academic environment, resources, and continuous support necessary for carrying out this research work. We are especially thankful to faculty members, and colleagues for their valuable suggestions, encouragement, and constructive feedback throughout the study.

Funding: The authors declare that no specific funding was received from any public, commercial, or non-profit funding agencies for conducting this research.

Conflict of Interest: The author declare that they have no conflict of interest.

Declaration of Artificial Intelligence (AI) Assistance Process: Artificial Intelligence (AI)-based tools, including OpenAI's ChatGPT, were utilized during the preparation of this manuscript to assist with language improvement, content organization, and grammar checking. The authors carefully reviewed, validated, and revised all AI-generated content and take full responsibility for the accuracy, originality, and integrity of the final manuscript.

Author Contributions: All author contributes equally to the research, design the study, analyzed data and approved the final manuscript

Ethics approval: This study does not involve human participants, animals, or any ethical concerns requiring approval from an ethics committee. Therefore, ethical approval was not required for this research.

Data availability: The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Abbreviations

BERT: Bidirectional Encoder Representations from Transformers, SBERT: Sentence BERT, TF-IDF: Term Frequency-Inverse Document Frequency, RMSE: Root Mean Square Error.

References

1. Burrows S, Gurevych I, Stein B. The eras and trends of automatic short answer grading. *Int J Artif Intell Educ.* 2015;25(1):60-117.
2. Mohler M, Bunesco R, Mihalcea R. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: *Proc 49th Annu Meet Assoc Comput Linguist*; 2011. p. 752-762.
3. Siddiqi R, Harrison CJ, Siddiqi R. Improving teaching and learning through automated short-answer marking. *IEEE Trans Learn Technol.* 2010;3(3):237-249.
4. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: *Proc Conf Empir Methods Nat Lang Process*; 2019. p. 3982-3992.
5. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y. BERTScore: evaluating text generation with BERT. In: *Proc Int Conf Learn Represent*; 2020.
6. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proc Conf North Am Chapter Assoc Comput Linguist*; 2019. p. 4171-4186.
7. Grootendorst M. KeyBERT: minimal keyword extraction with BERT. Zenodo; 2020.
8. Honnibal M, Montani I. spaCy 2: natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.
9. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: *Proc Int Conf Learn Represent*; 2013.
10. Dietterich TG. Ensemble methods in machine learning. In: *Proc Int Workshop Multiple Classifier Syst*; 2000. p. 1-15.
11. Zhou ZH. Ensemble methods: foundations and algorithms. Boca Raton: Chapman and Hall/CRC; 2012.
12. VanLehn K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ Psychol.* 2011;46(4):197-221.
13. Kulik JA, Fletcher JD. Effectiveness of intelligent tutoring systems: a meta-analytic review. *Rev Educ Res.* 2016;86(1):42-78.
14. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: *Proc Neural Inf Process Syst*; 2017. p. 5998-6008.
15. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: a robustly optimized BERT pretraining approach. *arXiv.* 2019;arXiv:1907.11692.
16. Rajpurkar P, Zhang J, Lopyrev K, Liang P. SQuAD: 100,000+ questions for machine comprehension of text. In: *Proc Conf Empir Methods Nat Lang Process*; 2016. p. 2383-2392.

17. Agirre E, Cer D, Diab M, Gonzalez-Agirre A. SemEval-2012 task 6: a pilot on semantic textual similarity. In: Proc Joint Conf Lexical Comput Semantics; 2012. p. 385-393.
18. Corbett AT, Anderson JR. Knowledge tracing: modeling the acquisition of procedural knowledge. User Model User-Adapt Interact. 1995;4(4):253-278.
19. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. In: Proc Neural Inf Process Syst; 2020. p. 1877-1901.
20. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(140):1-67.
21. Dzikovska M, Nielsen R, Brew C, Leacock C, Giampiccolo D, Bentivogli L, et al. Towards effective tutorial feedback for explanation questions: a dataset and baselines. In: Proc Conf North Am Chapter Assoc Comput Linguist; 2012. p. 200-210.
22. Ziai R, Ott N, Meurers D. Short answer assessment: establishing links between research strands. In: Proc Workshop Build Educ Appl NLP; 2012. p. 190-200.
23. Nguyen H, Xiong L. Classification of short answer grading via instance-based learning and ensemble methods. In: Proc IEEE Int Conf Data Min Workshops; 2018. p. 1180-1187.
24. Geetanjali, & Singh, A. R. (2025). Precision and smart farming: An approach to efficient resource management in agriculture. *Journal of Artificial Intelligence in Agriculture and Applied Research*, 1(1), 25–33. <https://agrpjournals.com/journals/journal/index.php/jaiaar/article/view/6>
25. Pandey, P. (2025). Artificial intelligence for early detection of dental caries from radiographic images: A machine learning. *Journal of Dentistry and Clinical Sciences*, 1(1), 31–39. <https://agrpjournals.com/journals/journal/index.php/jdacs/article/view/12>