Classification of Lecture Notes for Key Concept Coverage: A Comparison of Rule-Based, BERT, and LLM Approaches

Rohail Alam
rohail03@iastate.edu

Sahil Medepalli
sahilmed@iastate.edu

Advait Tilak
advait46@iastate.edu

1 Abstract

Assessing whether student notes capture key lecture concepts is an important yet challenging task in educational NLP. In this work, we address the problem of notes classification that is, predicting whether a student's notes contain a given lecture key point. We implement and compare three prescribed approaches. The rule-based model leveraging multiple rules preprocessed texts, achieved 72.2% accuracy and 57.99% F1 score, demonstrating the limitations of surface level methods. Our LLM-based approach utilizing Llama-4-Maverick through the Groq API shows improved performance (71.0% accuracy) but reveals a precision-recall tradeoff, with high recall (69.%) but lower precision (58.2%), suggesting tendencies toward over-prediction. The results highlight the trade-off between semantic understanding and operational constraints, with the rule-based method offering transparency but limited effectiveness, while the LLM provides better coverage at higher computational cost. Our BERT-based model achieved strongest performance (75.3% accuracy) by learning semantic relationships, offering a practical balance between accuracy and computational demands. These findings offer insights for implementing note-assessment systems in educational settings, where the choice of approach must balance accuracy, interpretability, and resource requirements.

2 Introduction

Effective note-taking is critical for learning, but manually evaluating whether students capture key lecture concepts is time-consuming. Automated grading/classification of notes captured can enable an easy feedback workflow for learners and instructors. However, this task poses unique challenges like semantic variability, partial or noisy notes etc.

Our rule-based approach that employs text preprocessing with a multi-tier positive and negative rule based architecture. This method achieves moderate performance (72.20% accuracy). But the approach's interpretability and computational efficiency come at the cost of reduced effectiveness and versatility when it comes to paraphrased or abstractly *(not explicitly)* related content.

The LLM approach demonstrates significantly improved accuracy over the rule-based model (71.0%) due to the model's semantic understanding capabilities. However, detailed analysis reveals important limitations: while recall reaches 69.7%, precision remains relatively low at 58.2%, indicating a tendency to over-predict coverage. This tradeoff, along with the solution's computational expense and API dependence, presents practical challenges for real-world deployment.

The BERT-based model fine-tuned on SciBERT, which demonstrates substantially improved performance (75.3% accuracy) by capturing deeper semantic relationships. This approach strikes a balance between the rule-based method's transparency and the LLM's sophisticated understanding, though it requires more computational resources for training than either alternative.

3 Related Work

Automated analysis of student generated content has been a central focus in the cross between NLP and education, particularly in assessing writing quality, identifying misconception, and content relevance.

Early work in concept coverage detection often relied on rule-based approaches, using approaches such as TF-IDF and Jaccard similarity to identify overlaps between student reponses and model answers (leacock et al., 2003). While interpretable and efficient, these approaches struggle with paraphrased expressions for example.

The advent of transformer based models has shifted the focus towards a deeper semanting understanding. BERT and its variants such as SciBERT (Beltagy et al. 2019) have been seen to be effective in educational contexts, especially when assessing question answer, short answers, and relevance (Riordan et al. 2019). These models excel in encoding contextual meaning and have outperformed prior baselines in tasks involving semantic inference.

Recently, LLMs have come under the spotlight in education, and have enabled flexible and generalizable approaches to text understanding. Techniques such as prompt engineering have shown promising results in many NLP tasks (Brown et al. 2020). In education, LLMs have been explored for feedback generation, automatic grading, ans summarization (Basu et al. 2023).

4 Methodology

4.1 Rule Based:

The primary goal was to make an interpretable, rule-based system to determine the quality of student notes. The system's design explicitly avoids statistical machine learning models, focusing instead on interpretable, configurable rules. To enable consistent comparison, both IdeaUnits and student notes underwent a sequential preprocessing pipeline. This involved removal of non-alphanumeric characters, lemmatization, tokenization and extraction of proper nouns using PoS Tagging.

The system employs several heuristic functions to assess the relationship between an IdeaUnit I and a note segment N:

1. Semantic Similarity: The semantic relatedness was assessed using spaCy's 'en_core_web_md' model. For a given I and N, their respective processed texts were converted into spaCy 'Doc' objects, and a semantic similarity score, $S(I, N)$, was computed using

$$S(I, N) \geq T_{sem} + \delta_S \tag{1}$$

where T_{sem} is a baseline semantic threshold, and δ_S is an adjustment factor depending on the context.

2. Lexical N-gram Overlap: The congruence at the n-gram level was measured. Let $G_k(T_p)$ be the set of k-grams derived from a set of preprocessed tokens T_p. The n-gram overlap ratio for k-grams is defined as:

$$R_{\text{ngram}}(I_p, N_p, k) = \frac{|G_k(I_p) \cap G_k(N_p)|}{|G_k(I_p)|}, \quad |G_k(I_p)| > 0 \tag{2}$$

A match component based on this is active if $R_{ngram}(I_p, N_p, k) \geq T_{ngram}$, where T_{ngram} is a predefined n-gram overlap threshold.

3. Proper Noun Concordance: A check was implemented for the presence of essential proper nouns. Here Levenshtein distance between proper noun from I and N is calculated and a maximum relative distance ratio (e.g., 0.2) is set to allow for minor misspellings.

4. Synonym-Enhanced Keyword Matching: This component assesses keyword presence, augmented by synonyms. Let K_I be the set of significant keywords extracted from I_p (after stop word removal). Let $Syn(kw)$ denote the set of synonyms for a keyword kw obtained via WordNet. The count of matched keywords is:

$$M_{syn} = \sum_{kw \in K_I} \mathbb{I}\left(kw \in N_p \vee (\exists s \in Syn(kw) \text{ s.t. } s \in N_p)\right) \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function (1 if true, 0 if false). The match is positive if $(M_{syn}/|K_I|) \geq T_{syn\_ratio}$, where T_{syn_ratio} is a minimum match ratio threshold.

The final classification, $P(I, N) \in \{0, 1\}$, is derived from a structured application of rules. Initially, negative filter conditions are applied. $P(I, N)$ is set to 0 if the preprocessed note N_p is too short (e.g., $|N_p| < N_{min\_len}$) or, optionally, if the proper noun concordance check $C_{PN}(PN_I, N_p)$ fails.

If the instance passes these filters, a positive classification ($P(I, N) = 1$) is made if any one of a set of disjunctive rule paths $(\Phi_1, \Phi_2, \ldots, \Phi_m)$ is satisfied:

$$\text{P(I,N) = 1} \iff (\text{NegativeFiltersPassed}) \wedge (\Phi_1 \vee \Phi_2 \vee \cdots \vee \Phi_m) \tag{4}$$

Each path Φ_j represents a distinct logical combination of the core matching components with specific thresholds. For example:

- Φ_{hc}: (High Confidence) Active if $S(I, N) \geq T_{sem\_strict}$ AND $R_{ngram}(I_p, N_p, k) \geq T_{ngram\_strict}$.
- Φ_{bal}: (Balanced) Active if $(S(I, N) \geq T_{sem\_mod}$ AND $R_{ngram}(I_p, N_p, k) \geq T_{ngram\_lenient})$ OR $(S(I, N) \geq T_{sem\_lenient}$ AND $R_{ngram}(I_p, N_p, k) \geq T_{ngram\_mod})$.
- Other paths focused on strong n-gram evidence, synonym matches, or special handling for short IdeaUnits, each with their own specific thresholds and logic.

This multi-path approach allows for flexibility in identifying concept coverage under varied textual evidence.

4.2 BERT:

To prepare the data, each entry was first linked to eachother from the training and testing datasets to its corresponding note in Notes.csv using the ID and Segment fields. Mapping it like this allowed for the resconstruction of the complete input for each example. The Note and the target IdeaUnit was then concatenated using the [SEP] token, allowing the model to treat the task as a form of sentence pair classfication, which is a common use case for a transformer based model like BERT. This way, the final input was formated as "Note text [SEP] IdeaUnit".

HuggingFace's BertTokenizer was used for tokenization, and the data was converted into the HuggingFace Dataset format as well. This allowed for integration with the Trainer API, which was used to fine tune the models. Each dataset was tokenized with truncation and padding applied to a maximum sequence length of 512 tokens to standardize inputs for training. Additionally, a CrossEntropyLoss function was employed which is the default loss when the model's final layer is a classification head. Multiple BERT variants were assessed to compare performance and to select the best model.

4.3 LLM:

```
You are given a set of shorthand lecture
 notes written in fragmented style. Your
 task is to convert these notes into a
single coherent paragraph using full,
grammatically correct sentences.

Instructions:
1. Do not add any information not found
in the notes.
2. Do not include explanations, labels,
or commentaryreturn only the final
paragraph.
3. Preserve all factual details and
express them in smooth, natural English.
4. The output must be a self-contained
paragraph suitable for use in a textbook
 or structured dataset.

Input:
(Selected Segment Note)
Output:
Only return the converted paragraph.
```

This approach employs a Large Language Model (LLM) to perform the core task, accessed via the Meta LLM API provided by Groq. Specifically, the Maverick variant of the recently introduced LLaMA-4 series is used. This variant has been reported to outperform its current competitors and represents the latest advancement within the LLaMA family. Its strong performance and modern architecture make it a suitable choice for this use case.

The ground truth dataset undergoes a refinement process in which notes are transformed into coherent and complete sentences. This is accomplished using a separate prompt for each segment within every lecture entry in the dataset. The prompt is structured as shown in the aforementioned prompt at the beginning of the section.

The input to the model is carefully designed using prompt engineering techniques to ensure precise and consistent responses. Each prompt is structured to frame the task and guide the model effectively. The task is introduced as an expert-level analysis of student notes from a lecture, aiming to determine whether each note captures a key concept referred to as an *IdeaUnit*.
To constrain the model's output and ensure it aligns with the expected format, additional instructions are appended to the prompt:

- **ONLY** output a Python-style array of Booleans (e.g., `[True, False, ...]`).
- Each Boolean in the array must correspond to the note at the same index.
- **DO NOT** provide any explanations or additional text.

This prompt format allows the model to reason based on prior examples and generate direct, interpretable outputs suitable for automated evaluation:

```
You are an expert in analyzing student
notes from a lecture to determine
whether they cover a specific key point,
 called an “IdeaUnit”.

Here is the example of an IdeaUnit and
some examples of student notes and
whether these notes cover the IdeaUnit
or not:

IdeaUnit: (IdeaUnit corresponding to the
 same ID and Segment Number as the test
sample)

Below are notes from different students.
 For each note, check if it covers the
IdeaUnit.

Example x:
Notes: (Student note for the lecture)
Covers the IdeaUnit?: (True/False)

... (Other examples follow)

Using your understanding from the above
examples, determine if the below
IdeaUnit is covered by the series of
notes which follows:

IdeaUnit: (IdeaUnit of the sample being
evaluated)

Note x: (Notes from student "x")
... (Remaining student notes for the
same ID and Segment Number)

For each note, check if it covers the
IdeaUnit. Answer with 'True' or 'False'.
```

5 Experiments

5.1 Rule Based:

The model's efficacy was assessed with standard classification metrics gauging performance on distinct training and test sets. During development, an iterative process was followed to optimize internal thresholds for the semantic, n-gram, and keyword matching components was employed to optimize the rule-based logic. The system demonstrated a reasonable ability correctly identify concept coverage in training examples. On unseen test data, while the model's positive predictions of concept coverage were largely trustworthy, its recall indicated a significant challenge in capturing the full breadth of relevant IdeaUnits. This tendency to miss existing concepts (false negatives) more often than incorrectly flagging non-existent ones (false positives) resulted in a moderate overall F1-score, underscoring the difficulty of creating universally applicable rules for diverse textual expressions.

The system demonstrated a reasonable ability correctly identify concept coverage in training examples. On unseen test data, while the model's positive predic-

Metric	Training Data	Testing Data
Accuracy	0.6627	0.7220
F1 Score	0.6532	0.5799
Precision	0.6532	0.6383
Recall	0.6532	0.5313

Table 1: Results for Rule base Model

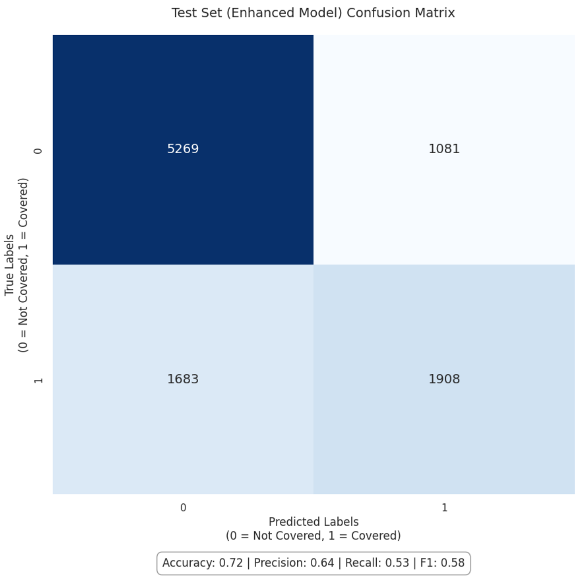

Figure 1: Confusion Matrix for Rule Based Model

tions of concept coverage were largely trustworthy, its recall indicated a significant challenge in capturing the full breadth of relevant IdeaUnits. 1 This tendency to miss existing concepts (false negatives) more often than incorrectly flagging non-existent ones (false positives) resulted in a moderate overall F1-score, underscoring the difficulty of creating universally applicable rules for diverse textual expressions.

5.2 BERT:

There were three models that were tested with this task: bert-base-uncased, roberta-base, and scibert-scivocab-uncased. These models were chosen based on their ability to understand general english, robust language understanding, and academic language understanding. Each model was fine tuned using a learning rate of 3e-5, a batch size of 16, and five epochs. These models were evaluated using accuracy, precision, recall, and F1 score. The results for these models are above.

Model	Accuracy	Precision	Recall	F1
bert-base-uncased	0.5882	0.4461	0.5798	0.5042
roberta-base	0.3612	0.3612	1.0000	0.5307
scibert_scivocab_uncased	0.7531	0.6708	0.6218	0.6454

Table 2: Results for BERT Models

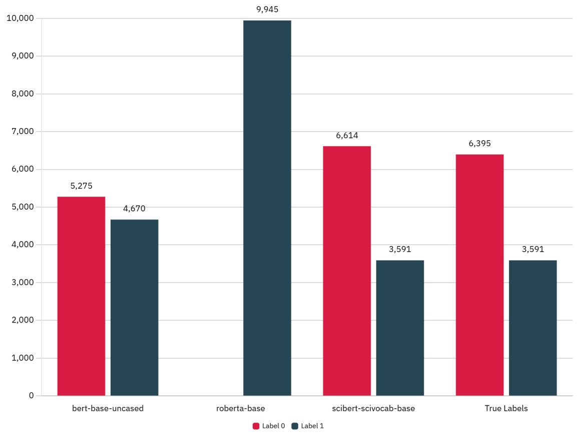

Figure 2: Distribution of Predicted Lables (BERT)

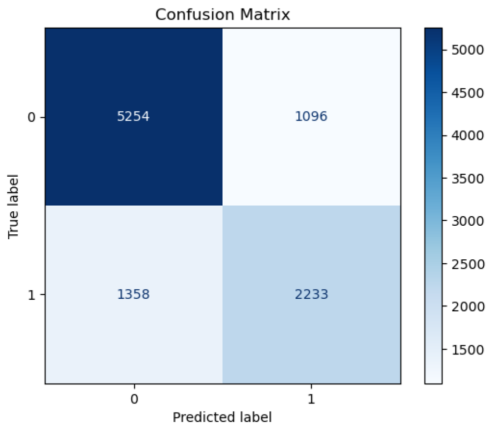

Figure 3: Confution Matrix for scibert-scivocab-uncased

Initial results using bert-base-uncased yielded average performance with an accuracy of .59 and F1 score of .50, however the model struggled with class imabalance, and heavily favored the majority class. roberta-base also struggled with this, and only predicted the positive class. scibert-scivocab-uncased performed the best, achieving almost .70 accuracy and 0.6 recall. Undersampling of the majority class was performed to address class imbalance since there were significantly more notes labeled 0 than 1. There was a randomly remobed subset of label 0 to balance the dataset. This resulted in improved recall, and was visualized in the final confusion matrix of scibert-scivocab-uncased, which shows an even distribution of predictions accross the classes, showing the model learned to recognize both labels.

5.3 LLM:

The predictions for the LLM were quite cumbersome to obtain, due to the sheer size of the testing dataset and hence the number of prompts required. It would be very inefficient to individually prompt each and every sample, which is why the prompts were given for batches of testing data according to the 'Topic', 'ID' and 'Segment' columns. Samples which had the same value for all three of these attributes were grouped together in the prompt. With the help of prompt engineering, the model output was consistent and therefore easy to infer, despite the slightly increased complexity of the prompt due to batching. Error handling through try catch statements was also implemented in the code to avoid interruptions due to temporary loss of network, read errors, or any other miscellaneous issue.

Metric	Value
Accuracy	0.7672
F1 Score	0.6764
Precision	0.6793
Recall	0.6736

Table 3: Metrics obtained from the test dataset

Table 3 does not include training data metrics for the LLM model, as it is not trained in the conventional sense. Instead, the training data is provided directly through the prompt. This approach allows the model to recognize patterns and infer the evaluation strategy dynamically, without undergoing parameter updates. Despite this lack of fine tuning, the LLM demonstrates performance that surpassed the rule-based baseline model as well as the BERT-based model, showcasing its ability to generalize from limited in-context examples.

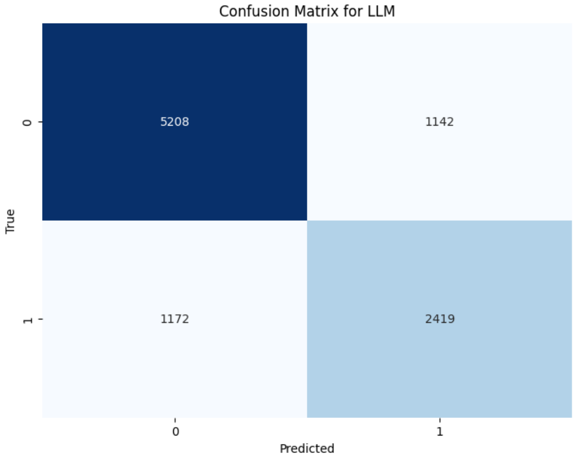

Figure 4: Confusion Matrix for LLM-based approach

6 Conclusion

This report explored the use of Rule-Based, BERT, and LLM approaches to automate feedback on lecture note taking. The Rule-Based model, which emphasized semantic similarity, lexical N-gram overlap, proper noun concordance, and synonym enhanced keyword matching, classified with an accuracy of .7220 on the testing set, with an F1 score of 0.5799, and was ultimately held back by the inherent rigidity of the approach, and inability to capture semantic nuances. This was improved on using BERT, specifically SciBERT, seeing a 0.7531 accuracy along with a 0.6454 F1 score. Although it did a better job in capturing contextual phrasing, the one shot nature of the task and class imbalance limited its ability to fully generalize across different styles of note taking. The LLM approach using the Meta LLM API from Groq, used along with carefully designed prompt engineering techniques sees further improvement, yielding an accuracy of 0.7672, and F1 score of 0.6764. These results are limited by the free-teir usage constraints, and potential hallucinations, which requires extra layers and computational power to account for.

7 Limitations

7.1 Rule Based:

The rule-based approach, is fundamentally limited by its inherent rigidity in adapting to diverse linguistic variations and complex semantic nuances. The manual effort required to create and maintain a comprehensive rule set that can cover a wide range of topics and expressions is substantial. Furthermore, the system's performance is tied to the completeness of its synonym and abbreviation resources, and its capacity to discern subtle paraphrasing or abstract conceptual links remains constrained. The model is sensitive to fine-tuned thresholds which means that its current rules may not generalize to new subjects without recalibration.

7.2 BERT:

Despite the consistent improvements, the approaches taken using BERT have several limitations. First, the dataset is imbalanced, with far more 0 labels and 1 labels. Although undersampling helped, it still reduced the volume of training data that is available, which limits generalizability and stability of the models.

Furthermore, the one shot nature of the task limits the model's ability to generalize across different styles of note taking. There is no diverse training set for each topic, and the model may overfit to specific phrasings found in the annotated example, and fail to generalize to other semantically equivalent expressions.

Finally, while scibert-scivocab-uncased performed the best, its effectiveness is still limited by the token length of 512. In some cases, this can lead to truncation when concatenating the full note and IdeaUnit, leading to a loss of information.

7.3 LLM:

The only impeding factor in using the LLM model is the limitations imposed by the free-tier usage constraints. These restrictions cap the number of prompts that can be processed daily, potentially forcing a tradeoff between the quantity of prompts and the quality of each prompt—an issue that would not arise under unrestricted access. Although the chances are very slim, the LLM also could possibly hallucinate its results, which requires and extra layer of checking the quality of the response and isolating the relevant information from it.

References

1. Basu, Sweta, et al. "Automated Feedback Generation in Education Using Large Language Models."

Journal of Educational Data Mining, vol. 15, no. 2, 2023, pp. 45–67.

2. Beltagy, Iz, Kyle Lo, and Arman Cohan. "SciBERT: A Pretrained Language Model for Scientific Text." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, edited by Anna Korhonen et al., Association for Computational Linguistics, 2019, pp. 3615–3620.

3. Brown, Tom B., et al. "Language Models Are Few-Shot Learners." Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.

4. Devlin, Jacob, 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, edited by Daniel Marcu and Katrin Erk, Association for Computational Linguistics, 2019, pp. 4171–4186.

5. Horbach, Sebastian, and Iryna Gurevych. "Detecting Misconceptions in Student Writing with Transformer Models." Proceedings of the 16th International Conference on Educational Data Mining, International Educational Data Mining Society, 2021, pp. 125–134.

6. Kojima, Tom, et al. "Large Language Models Are Zero-Shot Reasoners." arXiv, 2 Feb. 2022, arxiv.org/abs/2205.11916.

7. Leacock, Claudia, and Martin Chodorow. "C-Rater: Automated Scoring of Short-Answer Questions." Computers and the Humanities, vol. 42, no. 4, 2008, pp. 389–405.

8. Mohler, Mary, and Rada Mihalcea. "Text-to-Text Semantic Similarity for Automatic Short Answer Grading." Proceedings of the 22nd International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence, 2011, pp. 1093–1098.

9. Riordan, Ryan, Jacob Eisenstein, and Eduard Hovy. "Exploring BERT for Short Answer Grading in Educational Settings." Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, 2019, pp. 101–111.

10. "LLaMA 4 Maverick API Documentation." Groq, 2024, docs.groq.com/maverick-llama4.

11. "LLaMA 4: Technical Report." Meta Platforms, Inc., 2024.

12. Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.