A Reimplementation of Integrating ChatGPT into Secure Hospital Networks: A Case Study on Improving Radiology Report Analysis

Jacob Fuehne, Jared Backofen, Lokanath Das

Department of Computer Science, University of Illinois at Urbana-Champaign
Champaign, Illinois, USA
{jfuehne2, jaredb3, ldas2}@illinois.edu

Abstract

We reproduce and extend the study of Kim et al. (2024) on distilling a cloud-hosted large language model into on-premise student models for radiology report classification under strict privacy constraints. Using the MIMIC-CXR dataset, we reimplement their document-level knowledge distillation pipeline with a ternary label space (normal, abnormal, uncertain) and supervised contrastive learning. We replace the original ChatGPT-3.5 teacher with modern foundation models and add new student architectures such as BioClinicalBERT and DeBERTa. We also explore a mixture-of-experts teacher labeling strategy that aggregates multiple LLMs instead of using multi-shot prompting for a single model. Our results assess how reproducible the original findings are and how updated teachers and students behave in a privacy-preserving clinical NLP setting.

Project Video:
https://mediaspace.illinois.edu/media/t/1_h32zotxz
Project Github: https://github.com/CS-598-Deep-Learning-for-Healthcare/Onprem-Radiology-Anomaly-Classification/tree/main
PyHealth Pull Request:
https://github.com/sunlabuiuc/PyHealth/pull/639

Introduction

Hospitals face challenges when it comes to utilizing state-of-the-art large language models (LLMs) while also preserving patient privacy. Compliance requirements such as HIPAA and GDPR restrict hospitals from sending protected health information to third-party cloud-hosted LLM services where they do not have strict guarantees on data handling and retention. The paper by Kim et al. (2024) proposes getting around this limitation by transferring labels generated by a high-performance, insecure cloud LLM to a smaller, locally hosted and secure LLM model through knowledge distillation.

The original work focuses on classification of radiology reports from the MIMIC-CXR dataset into three categories: *normal*, *abnormal*, and *uncertain*. Their contributions include (1) comparison between sentence-level vs document-level knowledge distillation, (2) an explicit *uncertain* class to capture ambiguous or mixed findings, and (3) a supervised contrastive loss that improves separation between classes in the latent space. Student backbones such as RadBERT-RoBERTa, BioMed-RoBERTa, BioBERT, BlueBERT, and ClinicalBERT are evaluated using accuracy, sensitivity, specificity, AUC, and visualizations of the learned representation space.

Scope of Reproducibility

Our goal is to reproduce the document-level knowledge distillation and evaluate how robust the proposed methodology is using modern LLMs.

The authors did not release their teacher-generated labels or grant access to pretrained student checkpoints, and the exact version of ChatGPT-3.5 they used is unknown (and likely retired). Therefore, exact numerical reproduction is impossible. Instead, we aim for methodological reproducibility: we recreate the pipeline as closely as possible using the same dataset, similar model families, and identical metrics, and then compare trends and relative performance.

Methodology

Environment

All experiments were conducted in Python 3.10, using the PyTorch and HuggingFace transformers libraries together with Databricks for data access and batch LLM labeling.

Python and Dependencies Core packages include:

- `torch` for model training and GPU acceleration.
- `transformers` for loading Huggingface models.
- `datasets`, `pandas`, and `numpy` for data processing.
- `scikit-learn` for evaluation metrics.
- `matplotlib` for plotting latent-space visualizations.

We also used the Databricks SQL Warehouse client and Foundational LLM serving endpoint APIs.

Data

Dataset Access and Preparation We use the MIMIC-CXR dataset (Johnson et al. 2019), a large, publicly available collection of chest radiographs and associated text reports. Access requires completing MIT CITI training and signing a data use agreement. The dataset is available at:

https://physionet.org/content/mimic-cxr/2.1.0/

We follow the original paper by focusing on the text reports, in particular the *Findings* and *Impression* sections.

Train / Test Splits We created our splits following the original papers setup:

- **Training set:** Findings and impressions from all reports in partition p10, resulting in 22,197 samples.
- **Test set:** The first 5,217 findings and impressions from reports in partition p11.

Each example is associated with `subject_id`, `study_id`, findings text, and impression text.

Teacher Labels The dataset does not ship with anomaly labels. Instead, following Kim et al. (2024), we generate labels using cloud-hosted LLM teacher models. Each example is assigned one of: *Normal*, *Abnormal*, *Uncertain*.

After label generation, we apply post-processing to remove low-confidence or *Uncertain* examples for student training.

LLM Usage for Data Preprocessing Teacher labels and explanations are generated via Databricks-hosted LLM endpoints. The following prompt was used:

> Given the input as findings and impression, output ONLY a valid and strict JSON object within {} with two fields: `label` (a single word: Normal, Abnormal, or Uncertain) and `explanation` (a detailed reason matching the label). Do not include any other text. Findings: [FINDINGS]. Impression: [IMPRESSION].

The resulting labels and explanations appeared consistent with the underlying reports and are returned in well-structured JSON, which makes parsing straightforward. Because we lack human-annotated ground truth, we rely on manual checks for validation and on downstream student performance as an indirect signal of label quality. The same prompt template is reused for all teacher models and for both training and test sets.

Model

Original Repository The original implementation is available at:

https://github.com/Junhyun-Park01/OnPremise-Radiology-Anomaly-Classification

We use this repository as a starting point and adapt it to our environment, data source, and model choices.

Teacher–Student Architecture The overall setup follows a teacher–student distillation pattern:

- **Teacher:** A cloud-hosted LLM generates document-level ternary labels and explanations from the raw findings and impression text.
- **Student:** A smaller encoder model is fine-tuned on the teacher-generated labels to run locally inside a secure environment.

In the original work, the teacher is ChatGPT-3.5, and student backbones include RadBERT-RoBERTa, BioMed-RoBERTa, BioBERT, BlueBERT, and ClinicalBERT. In our reproduction and extensions:

- We replace the teacher with three modern models: GPT 5.1, Llama4 Maverick, and Qwen3 Next 80B.
- We keep RadBERT as a baseline student and add Bio-ClinicalBERT (pretrained on MIMIC-III (Johnson et al. 2016)) and DeBERTaV3 (He, Gao, and Chen 2021) as additional student models.

Student models take as input a radiology report and output a class in {Normal, Abnormal}. *Uncertain* labels were filtered out as part of the high-confidence label extraction following the original paper. We train three variants. First, a supervised baseline using standard cross-entropy loss. Second, we train an encoder with supervised contrastive loss on document embedding. Third, we freeze the trained encoder and train a MLP classifier on top using cross-entropy loss.

LLM Assistance for Model Implementation Most of the training loop and model definition existed in the original repo, but we needed to:

- Replace `BertForSequenceClassification` with `AutoModelForSequenceClassification` to support additional backbones like DeBERTa.
- Integrate Databricks as the source of training and test data.
- Debug NaN values arising in the contrastive loss computation.

We used an LLM to help write Databricks SQL helper functions, given our warehouse URL, table names, and library versions. The LLM also assisted in diagnosing NaN losses: after several iterations where we provided the training code and custom loss function, it suggested normalizing the feature vectors used by the contrastive loss. Once we implemented this normalization, the NaNs were gone. This required multiple prompts and careful checking, but the LLM was helpful as a debugging assistant.

Pretrained Models Both teacher and student models are initialized from pretrained checkpoints:

- Teachers: GPT 5.1, Llama4 Maverick, and Qwen3 Next 80B via Databricks model serving.
- Students: RadBERT, BioClinicalBERT, and DeBERTaV3 loaded from HuggingFace.

We perform supervised fine-tuning on the teacher-generated labels.

Training

Hyperparameters We follow the general hyperparameter settings of the original paper. Hyperparameter values for all training runs:

- **Learning rate**: 4×10^{-5}
- **Batch size**: 16.
- **Number of epochs**: 20
- **Optimizer**: AdamW.
- **Sequence truncation length**: 120 tokens.
- **Contrastive temperature**: 0.07 for the supervised contrastive loss.
- **Pooling strategy**: Mean pooling over the last hidden state.

Computational Requirements We trained student models using a NVIDIA 5080 GPU. Training was manageable on a single local GPU, but took close to an hour to train all three student models.

Training Details and LLM Assistance We use AdamW with learning-rate scheduling and gradient clipping. An LLM helped integrate metric computation into the training loop and suggested code structure changes to keep the training, evaluation, and logging logic modular.

Evaluation

Metrics We use the same evaluation metrics as the original paper:

- **Accuracy**: $(TP + TN)/(TP + TN + FP + FN)$.
- **Sensitivity**: $TP/(TP + FN)$.
- **Specificity**: $TN/(TN + FP)$.
- **F1-score**: $2 \cdot \frac{\text{Precision} \cdot \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}$.
- **Latent-space visualization**: 3D t-SNE projections of document embeddings colored by class.

These metrics are standard in classification tasks and enable direct comparison to the results reported by Kim *et al.* Latent-space visualizations offer a qualitative view of how well the model separates normal vs. abnormal documents.

LLM Assistance for Evaluation The original training code already computed basic metrics; we used an LLM to help:

- Integrate accuracy, specificity, sensitivity, and F1 into the evaluation loop using `scikit-learn`.
- Add code to save metrics to CSV files for easier comparison across runs.
- Build a t-SNE–based latent-space visualization that roughly matches the figure presented in the paper.

Because the paper did not provide detailed implementation notes for their visualization, we supplied the LLM with an image of the original figure and asked it to generate similar plotting code. This approach saved time and produced qualitatively similar plots.

Results

Original Reported Results (Kim *et al.*)

Table 1 summarizes the document-level anomaly detection performance reported by Kim et al. (2024) across several student models. These results serve as the primary quantitative reference for our reproduction.

Reproduction Results

Table 2 reports the document-level results from our reproduction experiments, comparing standard supervised fine-tuning (baseline) against supervised contrastive pretraining followed by an MLP classifier. All results are computed on the same MIMIC-CXR test split used for testing.

Table 1: Document-level anomaly detection performance across various student models (reported by Kim *et al.*).

Model	Accuracy	Specificity	Sensitivity	AUC
RadBERT-Roberta -4m-document	85.52	85.8	84.0	90.1
BioMed-Roberta -document	86.12	82.0	86.9	87.7
BlueBERT -document	91.17	91.0	**92.2**	95.8
Clinical BERT -document	90.15	88.8	96.1	**96.8**
BiomedBERT -document	**92.76**	**93.0**	91.6	92.6
BioBERT -document	90.50	90.5	90.6	95.9

Comparison to the Original Paper

Because the original paper used teacher models such as older versions of ChatGPT 3.5 and did not make available their labeled MIMIC data, it was impossible to fully replicate the original paper for a one to one comparison, as we can't test on their same test set or train with their teacher model. Thus, our ablations and changes are necessary in the reproduction itself and can be found in Table 2. We used RadBERT-Roberta-4m-document and trained using their code with the same loss and the same number of epochs (with some fixes to some normalization errors).

LLM-Suggested Ideas

To decide on extensions, we prompted an LLM with our project goals and provided a copy of the original paper. The LLM suggested multiple ideas:

- Trying alternative loss functions and regularization techniques.
- Exploring binary vs. ternary labeling strategies.
- Using newer teacher and student models.
- Building ensembles or mixtures of teacher models.

We verified these ideas manually. One important lesson was that the LLM confidently suggested a binary-label ablation that turned out to already exist in the original codebase. We only caught this after carefully reading through the github implementation. This highlighted the need to treat LLM brainstorming as only a starting point.

Chosen Ablations: New Teachers, New Students, and Mixture-of-Experts

We selected three ablations:

- **New teacher models:** Replace ChatGPT-3.5 with ChatGPT 5.1, Llama4 Maverick, and Qwen3 Next 80B.
- **New student models:** Add BioClinicalBERT and DeBERTaV3 as student models.
- **Mixture-of-Experts ensembling:** Combine teacher outputs via a mixture-of-experts strategy.

Our hypothesis is that modern foundation models provide better labels than the original ChatGPT-3.5 and that combining multiple teachers will produce more robust labels than

relying on multi-shot sampling from a single model. We also expect BioClinicalBERT and DeBERTaV3 to be competitive with or better than RadBERT when trained with the same distillation setup.

Implementation of the Ablations

The mixture-of-experts teacher labeling was implemented in Databricks:

- For each example, we sent the same prompt to three teacher models (GPT 5.1, Llama4 Maverick, Qwen3 Next 80B).
- We used Databricks's `ai_query` feature to compute a confidence score or similarity for each teacher response.
- We treated each teacher output as a trial and applied the same accept/reject post-processing logic as Kim et al. (2024), selecting a single final label where the ensemble met confidence criteria and discarding low-confidence or *Uncertain* examples.

The new student models were integrated by:

- Switching to `AutoModelForSequenceClassificatior` in the training code.
- Loading the appropriate tokenizer and model weights for BioClinicalBERT and DeBERTaV3.
- Ensuring that the contrastive loss worked with their embedding sizes.

LLMs assisted with the refactoring to support multiple models, as well as with debugging a data normalization issue.

Ablation Results and Visualization

Table 2: Performance comparison between baseline and contrastive loss (New Data). **Bold** indicates the best performance between the two methods for a given backbone.

Model	Accuracy	Specificity	Sensitivity	F1
Bio_ClinicalBERT -document -baseline	**95.62**	89.34	**98.25**	**96.93**
Bio_ClinicalBERT -document -contrastive	95.57 (- 0.05)	**92.97** (+ 3.63)	96.67 (- 1.58)	96.85 (- 0.08)
DeBERTa-v3-base -document -baseline	**96.15**	**93.88**	97.11	**97.26**
DeBERTa-v3-base -document -contrastive	96.11 (- 0.04)	92.59 (- 1.29)	**97.59** (+ 0.48)	97.25 (- 0.01)
RadBERT-4m -document -baseline	**96.38**	**93.95**	97.40	**97.43**
RadBERT-4m -document -contrastive	96.29 (- 0.09)	93.50 (- 0.45)	**97.46** (+ 0.06)	97.37 (- 0.06)

Overall, there were consistent large gains in accuracy compared to the original paper from Kim et al. (2024) observed by implementing the teacher student methodology

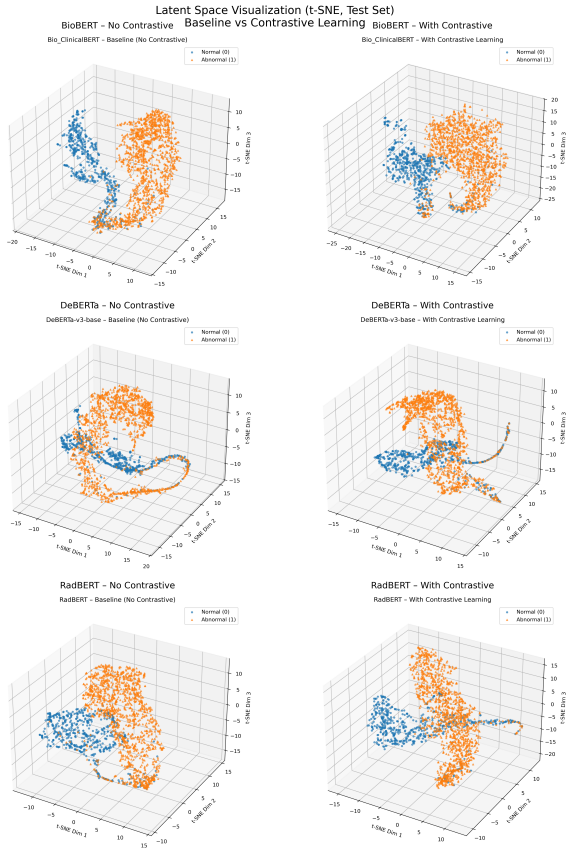

Figure 1: t-SNE visualization of document-level embeddings for BioClinicalBERT (top), DeBERTa-v3-base (middle), and RadBERT (bottom). Left panels show baseline training without contrastive loss, and right panels show training with supervised contrastive loss. Blue points indicate normal reports and orange points indicate abnormal reports.

with the latest SOTA LLM models, with a gain of approximately 10.1% on RadBERT-Roberta on document level anomaly detection. Based on our results in comparison to the original paper, we believe that contrastive loss training may not be necessary or worth the extra training time. There were consistent, but marginal (¡0.1%) losses in accuracy across all models tested at 20 epochs of training. There were also gains in performance of other metrics, but those gains are not consistently observed across models and the ranges of improvements may well be within the margin of error created by variations in test sets.

Discussion

Implications of the Experimental Results

Our results support the original paper's findings that one can use an ensemble based approach to ensure consistency between model generations to establish a ground truth of synthetic labels for training data, which can then be used to fine

tune a much smaller model that can be run on local hardware for uses such as a secure hospital network. By rejecting uncertain labels to remove edge cases, the labels can be limited to only examples where the teacher model would be able to resolve and greatly reduce the chances of our ground truth labels being different from a real ground truth by humans, though this does reduce the complexity of the resulting dataset. Thus, the metrics of the models are a rough approximation of their accuracy on real world scenarios, but could be more accurately interpreted to be how close the student model is to replicating the teacher models. From implementing this paper, we learned how challenging it can be to perform peer-review and reimplementations in machine learning healthcare applications due to the protected nature of data. Even in cases where the data is made public under certain restrictions, such as MIMIC, and you apply to get access under those same restrictions, authors often might not share their extensions to those datasets such as labeling, or the resulting trained models from that data. There is also a risk in doing research on cloud hosted models that are not open sourced such as ChatGPT 3.5, as these models will often become retired eventually, and in the case of ChatGPT 3.5, it's untestable in the exact form of this paper in less than 2 years. In such cases, it's impossible to perform direct comparisons of improvements for multiple reasons. Nevertheless, the results are improved so significantly from the original paper that it can be concluded that using our extensions will result in better performance.

What Was Easy

Accessing MIMIC-CXR and running baseline student models from HuggingFace were relatively straightforward. The original repository was reasonably organized, which reduced the engineering overhead for setting up initial experiments.

What Was Difficult

The most challenging aspects were:

- Reconstructing teacher labeling without the original ChatGPT-3.5 model or its prompts and parameters.
- Handling the absence of released labels and checkpoints, which prevented direct numerical comparison.
- Debugging NaN issues in the contrastive loss and ensuring consistent normalization.

Recommendations to the Original Authors

To improve reproducibility, we would recommend that future work:

- Release teacher-generated labels (or at least a subset).
- Document the exact versions used for cloud-hosted teacher LLMs.
- Provide access to trained student checkpoints and logs of training runs.

These would help in reproducing and extending research.

Author Contributions

- Lokanath Das led data preprocessing, including MIMIC-CXR extraction, Databricks integration, label and extract generation and MIMIC CXR dataset contribution to Pyhealth repository.
- Jacob Fuehne focused on model training, evaluation, and latent-space visualizations.
- Jared Backofen coordinated the reproducibility plan, designed and implemented the extensions, integrated LLM assistance into the workflow, and organized the manuscript.
- All authors contributed to analysis, writing and presentation.

References

Kim, K., Park, J., Langarica, S., Alkhadrawi, A. M., & Do, S. (2024). Integrating ChatGPT into Secure Hospital Networks: A Case Study on Improving Radiology Report Analysis. *CHIL 2024 / arXiv:2402.09358 [cs.AI]*.

Johnson, A. E. W., et al. (2019). MIMIC-CXR: A large publicly available database of labeled chest radiographs. *arXiv:1901.07042*.

He, P., Gao, J., & Chen, W. (2021). DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *arXiv:2111.09543*.

Johnson, A. E. W., Pollard, T. J., Shen, L., et al. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*.

Appendix A. Data Setup Details

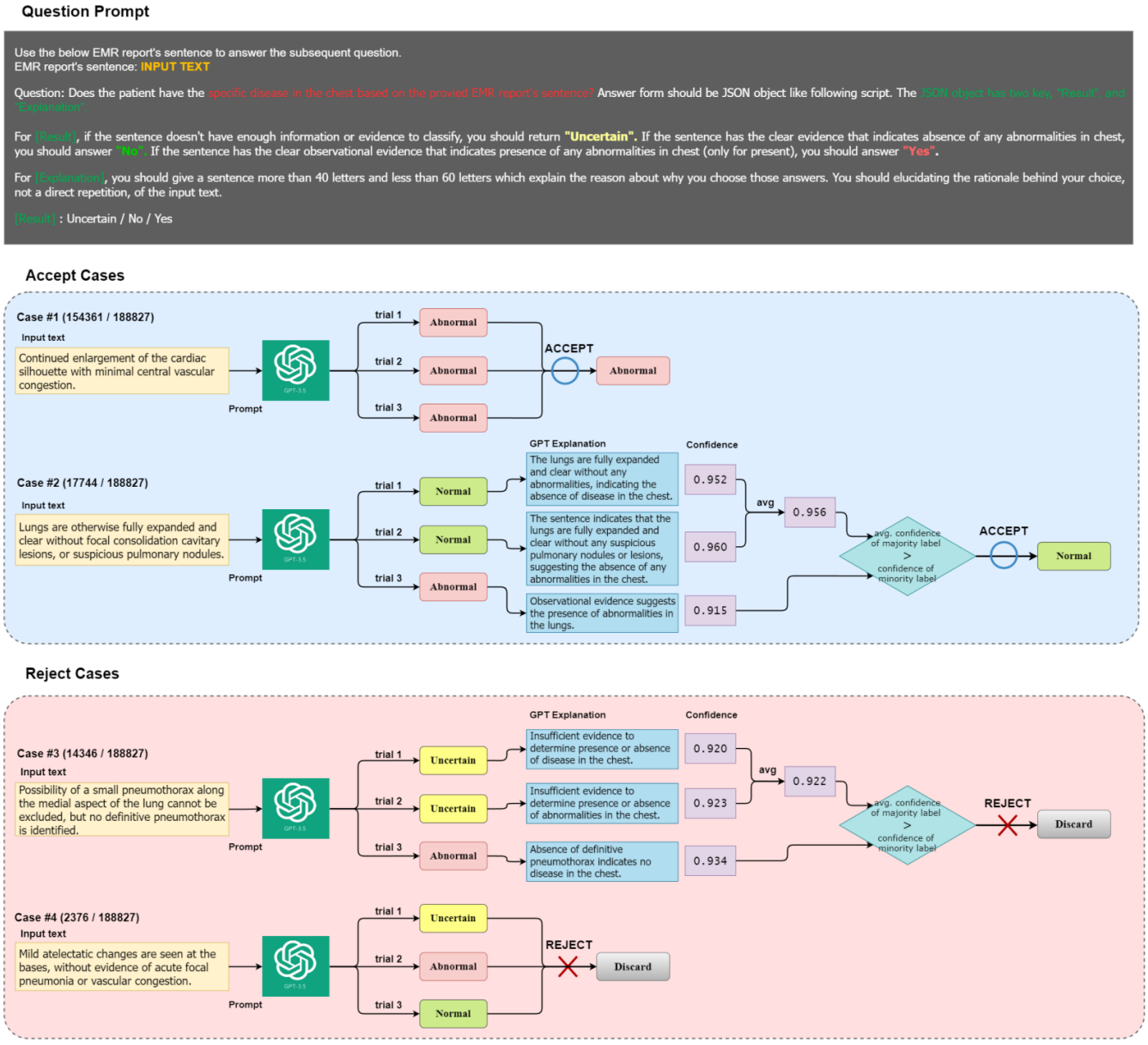

Figure 2: Kim et al.'s approach employs two key techniques to derive high-confidence sentence labels from GPT-3.5: (1) the prompt engineering that obliges the network to elucidate the rationale behind its outputs, and (2) the prediction ensemble methodology to extract the result ensuring all models concur in providing identical reasoning. Further explanation can be found in the appendix of Kim et al. (2024)