Maven at MEDIQA-CORR 2024: Leveraging RAG and Medical LLM for Error Detection and Correction in Medical Notes

Suramya Jadhav* , **Abhay Shanbhag*** , **Sumedh Joshi*** , **Atharva Date*** , **Sheetal Sonawane**
SCTR's Pune Institute of Computer Technology
{2018suramyajadhav,abhayshanbhag0110,sumedhjoshi463,atharva2718}@gmail.com,
sssonawane@pict.edu

Abstract

Addressing the critical challenge of identifying and rectifying medical errors in clinical notes, we present a novel approach tailored for the MEDIQA-CORR task @ NAACL-ClinicalNLP 2024, which comprises three subtasks: binary classification, span identification, and natural language generation for error detection and correction. Binary classification involves detecting whether the text contains a medical error; span identification entails identifying the text span associated with any detected error; and natural language generation focuses on providing a free text correction if a medical error exists. Our proposed architecture leverages Named Entity Recognition (NER) for identifying disease-related terms, Retrieval-Augmented Generation (RAG) for contextual understanding from external datasets, and a quantized and fine-tuned Palmyra model for error correction. Our model achieved a global rank of **5** with an aggregate score of **0.73298**, calculated as the mean of ROUGE-1-F, BERTScore, and BLEURT scores.

1 Introduction

Clinical notes typically include details about the patient's medical history, symptoms, physical examinations, diagnostic tests, treatments administered, and any other relevant information related to the patient's health status and care plan.

Accurate documentation is crucial for patient care, as errors in clinical notes can lead to misdiagnosis, improper treatment, and potential harm to patients. By automating the process of error detection and correction, healthcare providers can ensure the integrity and reliability of patient records, ultimately improving the quality of care delivered. Research indicates that a substantial proportion of adverse events in healthcare settings are due to errors in documentation, highlighting the need for effective error detection and correction mechanisms.

In this task of Medical Error Detection Correction Ben Abacha et al., 2024. We seek to address the problem of identifying and correcting medical errors in clinical notes. This task had 3 subtasks. In subtask 1 (Binary Classification) researchers had to detect whether the clinical notes included a medical error or not. Subtask 2 named Span Identification was to identify the text span associated with the error if a medical error exists. Subtask 3 (Natural Language Generation) was specifically to provide error-free text after making corrections if a medical error exists.

In our approach, we initially conducted Named Entity Recognition (NER) using GEMINI to identify words representing diseases or pathogens or suggestions in the text. After masking these identified words, we implemented the Retrieval-Augmented Generation (RAG) model on textbooks and external datasets. If the RAG score fell below a certain threshold, we passed the input to our model, which was made by using 4-bit quantization on Palmyra 20b and then fine-tuned the quantized Palmyra model using the QLoRA technique on MEDQA data Jin et al., 2020. If the word provided by Palmyra or the RAG model matched the word detected by NER, no error was detected. Otherwise, if a different word was obtained, it was replaced with the masked word identified by NER. Finally, the error sentence is mapped with the sentence ID to get the output in the desired format. This approach helped us in getting a Global Rank 5 with an Aggregate Score of 0.73298. The Aggregate score is calculated as the mean of ROUGE-1-F, BERTScore, and BLEURT . Our model achieved R1F, BERTSCORE, and BLEURT scores of 0.70306, 0.74372 and 0.75217 respectively.

The rest of the paper is organized as follows: Review of related work and background information in Sections 2 and 3 respectively, to provide context for our study. Following this, we elucidate the system architecture in Section 4 and describe the experimental setup in Section 5. Subsequently, we present our findings in Section 6, discuss limitations encountered in Section 7, and propose avenues for future research in Section 8. Finally, we have concluded our discussion in Section 9.

2 Background

The med dataset provided by organizers had 2 types of clinical notes - MS and UW. Upon meticulous examination of the datasets , it became clear that the medical dataset which was divided into MS and UW clinical notes presented some unique difficulties. The MS sub-

*first author, equal contribution

Proceedings of the 6th Clinical Natural Language Processing Workshop, pages 374–381

	MS	UW
Text ID	ms-val-108	uw-val-51
Text	A 3175-g (7-lb) female newborn is delivered at term. Initial examination shows a flat perineal Colonic atresia is confirmed when dark green discharge is coming out of the vulva.	Mr. <NAME/> has been noted to have documentation of thrombocytopenia on <DATE/> in the Medicine note. Plt 101 on admission. Thrombocytopenia was present on admission (POA).
Sentences	0 A 3175-g 1 (7-lb) female newborn is delivered at term. 2 Initial examination shows a .... 3 Colonic atresia is confirmed ...	0 Mr. <NAME/> has been noted to have documentation of thrombocytopenia on <DATE/> in the Medicine note. 1 Plt 101 on admission. 2 Thrombocytopenia was present on admission (POA).
Error Flag	1	0
Error Sentence ID	3	-1
Error Sentence	Colonic atresia is confirmed...	NA
Corrected Sentence	Imperforate anus is confirmed when dark green discharge is coming out of the vulva.	NA
Corrected Text	A 3175-g (7-lb) female ... opening. Imperforate anus is confirmed when dark green discharge is coming out of the vulva.	NA

Table 1: Dataset Glimpse

set, which came from Microsoft, had incredibly small flaws. So much so that a great deal of faults appeared to be subtle, making it difficult for the physicians on our team to recognize them. Yet, it was clear from closely examining the training set's corrected text that the corrections frequently represented ideal completions.

The UW subset, which came from University of Washington, on the other hand, showed a distinct scene. This subset of clinical notes seemed to more closely resemble real-world situations, which made errors easier to identify in them.

MS dataset was split into train (2189) and val (574), and UW into val dataset (160). The testing data was a mixture of MS and UW formats.

The dataset is in CSV format and consists of labeled text data. Each row represents a unique input text and includes columns named Text ID, Text, Sentences, Error Flag, Error Sentence ID, Error Sentence, Corrected Sentence, and Corrected Text. The Text column contains the complete text, while the Sentences column divides the text into individual sentences with corresponding IDs starting from 0. The Error Flag column indicates whether there is an error in the text, with 0 representing no error and 1 representing an error. If there is an error, the Error Sentence ID column specifies the ID of the sentence containing the error, and the Error Sentence column provides the erroneous part of the text containing the error. The Corrected Sentence column contains the error-corrected version of the sentence, and the Corrected Text column includes the complete text with corrected sentences. When there is no error, Error Flag is 0, Error Sentence ID is -1, and the Error Sentence, Corrected Sentence, and Corrected Text columns contain "NA" values. This structured format facilitates error detection and correction tasks within the dataset. Table 1 offers a glimpse into MS and UW datasets.

The MEDQA dataset is a collection of question-answer pairs related to the medical field specifically derived from professional medical board exams, like the United States Medical Licensing Examination (USMLE). It covers a wide range of medical topics and is available in three languages: English, Simplified Chinese, and Traditional Chinese.

Question-Answer Pairs: The dataset consists of multiple-choice questions along with their corresponding answers. The number of questions varies depending on the language:
English: 12,723 questions
Simplified Chinese: 34,251 questions
Traditional Chinese: 14,123 questions

Medical Textbooks: The dataset also provides access to a large corpus of medical textbook content to aid models in comprehending the medical context for answering the questions.

For this task we used the just the English QA corpus. Here's an example of a question-answer pair in MEDQA dataset.

Question A 55-year-old female patient presents with a chief complaint of progressive shortness of breath over the past 6 months. She denies chest pain, cough, fever, or chills. On physical exam, her vital signs are normal. Her lungs are clear to auscultation bilaterally.What is the most likely diagnosis for this patient's shortness of breath?

Options
A. Heart failure
B. Asthma
C. Chronic obstructive pulmonary disease (COPD)

D. Pneumonia
Answer-idx : C

3 Related Work

Zhu et al., 2024 unveils REALM, a Retrieval-Augmented Generation framework, addressing limitations in existing clinical predictive models by enhancing multimodal Electronic Health Records (EHR) representations. Integrating clinical notes and time-series EHR data, REALM leverages Large Language Models (LLM) and GRU models for encoding, while incorporating external knowledge from a labeled knowledge graph (PrimeKG). By aligning with clinical standards, the framework eliminates hallucinations and ensures consistency, culminating in an adaptive multimodal fusion network. Extensive experiments on MIMICIII tasks demonstrate REALM's superior performance, highlighting its effectiveness in refining multimodal EHR data utilization and enhancing nuanced medical context for informed clinical predictions.

Elgedawy et al., 2024 presented a conversational interface powered by large language models (LLMs) for efficiently accessing information within clinical notes. Utilizing Langchain framework and transformer-based models, users can interactively query and retrieve relevant details from unstructured clinical data. Evaluation experiments, including advanced language models and semantic embedding techniques, demonstrate promising results, with Wizard Vicuna showing the highest accuracy despite computational demands. Model optimization techniques, such as weight quantization, significantly improve inference latency. However, challenges like model hallucinations and limited evaluation across diverse medical cases remain, indicating avenues for future research in enhancing clinical decision-making through AI-driven approaches.

Singhal et al., 2023 outlines Med-PaLM 2, a significant advancement in medical question answering, achieving an impressive accuracy of 86.5 % on the MedQA dataset. Compared to its predecessor, Med-PaLM, which scored 67.2% on the same dataset, Med-PaLM 2 represents a substantial improvement. By leveraging enhancements in base large language models (LLMs), domain-specific fine-tuning, and novel prompting strategies, Med-PaLM 2 demonstrates promising progress towards attaining physician-level performance in medical question answering across various datasets, including MedQA, PubmedQA Jin et al., 2019, MMLU, and MedMCQA Pal et al., 2022.

Jin et al., 2020 elucidates MEDQA, the inaugural free-form multiple-choice OpenQA dataset for medical problem-solving, sourced from professional medical board exams in English, simplified Chinese, and traditional Chinese. With question counts of 12,723, 34,251, and 14,123 across the three languages respectively, MEDQA provides a robust benchmark. Despite employing both rule-based and neural methods, even the most advanced model achieves only 36.7%, 42.0%, and 70.1% test accuracy on English, traditional Chinese, and simplified Chinese questions. MEDQA poses significant challenges to current OpenQA systems, encouraging the NLP community to develop more robust models for medical applications.

Chen et al., 2023 introduces MEDITRON, an open-source suite of Large Language Models (LLMs) tailored for the medical domain, ranging from 7B to 70B parameters. Leveraging Nvidia's Megatron-LM Shoeybi et al., 2020 distributed trainer and a carefully curated medical corpus, including PubMed articles and international medical guidelines, MEDITRON outperforms state-of-the-art baselines across four major medical benchmarks. The study underscores the impact of increasing model parameters on medical LLM performance, highlighting MEDITRON's competitive edge against closed-source counterparts like GPT-3.5 and Med-PaLM. Notably, MEDITRON achieves performance levels within 5% of GPT-4 and 10% of Med-PaLM-2, thus potentially democratizing access to extensive medical knowledge.

The recent development of LLMs Boiko et al., 2023,Tamkin et al., 2021 has generated a great deal of enthusiasm due to their exceptional performance in natural language generation and understanding, as well as their adaptability in handling a variety of tasks. To improve the performance of Large Language Models (LLMs), particularly for disease identification and classification tasks. Oniani et al., 2024 proposed an ensemble prompting method called Models-Vote Prompting (MVP). The way MVP operates is that multiple LLMs are given the same task, and their results are combined via a majority voting procedure. The utility of MVP is demonstrated by experiments showing better results on one-shot unusual disease diagnostic tasks compared to distinct models in the ensemble. Additionally, the researchers provide a novel rare disease dataset, which is made available to researchers under the terms of the MIMIC-IV Data Use Agreement (DUA). For doing research and evaluating in the field, this set of data is a helpful resource.

The Retrieval Augmented Generation (RAG) Lewis et al., 2020 method is a natural language processing model that combines retrieval and generation components to handle knowledge-intensive tasks. In this paper Jin et al., 2024 used LLMs along with RAG to evaluate health reports with a novel feature extraction method. They used RAG to retrieve knowledge from the professional knowledge base. Researchers employ an automated feature engineering approach to train a classification model XGBoost for final disease prediction. The accuracy of GPT-4 combined with information retrieval by RAG for disease diagnosis is 0.68, and the F1 score is 0.71, while their framework achieved an accuracy of 0.833 and an F1 score of 0.762, respectively.

Dettmers et al., 2023 formerly employed QLoRA, an effective finetuning technique that maintained full 16-bit fine-tuning task performance while reducing memory usage to the point where a single 48GB GPU could finetune a 65B parameter model. The Guanaco model

family, described in the research as the top model family, achieves 99.3% of ChatGPT's performance level on the Vicuna test, beating out all other publicly available models in under 24 hours of fine-tuning on just one GPU. Results from this approach consistently demonstrate that, on educational standards with widely recognized evaluation settings, 4-bit QLORA with NF4 data type matches 16-bit complete finetuning and 16-bit LoRA finetuning performance. Additionally, they have demonstrated that NF4 (4-bit NormalFloat) outperforms FP4 (4-bit Float) and even indicated that performance is not diminished by double quantization.

A significant advancement in the field has been made recently with the development of HEAL Yuan et al., 2024, a Large Language Model (LLM) designed specifically for automated scribing and medical conversations. Based on the widely taught 13B LLaMA2 architecture, HEAL provides a novel approach to the unique issues associated with medical communication. An evaluation of HEAL on tasks like PubMedQA yields an excellent accuracy of 78.4%, proving its superiority over current LLMs like GPT-4 and PMC-LLaMA Wu et al., 2023. Furthermore, when it comes to producing medical notes, HEAL performs similarly to GPT-4, demonstrating its effectiveness in clinical documentation activities. Notably, HEAL outperforms human scribes and other similar models in terms of accuracy and completeness, and it outperforms GPT-4 and Med-PaLM 2 in terms of reliably identifying medical ideas.

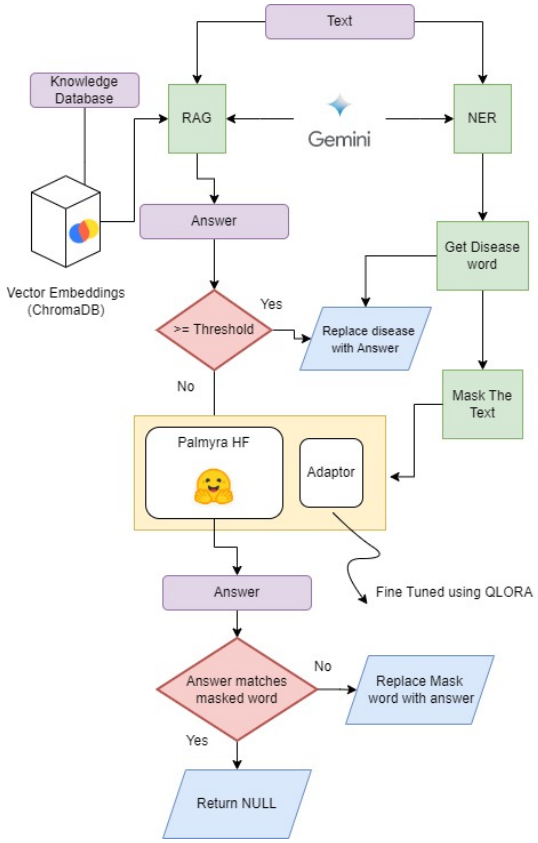

Figure 1: Proposed Model - Quantised Palmyra with RAG

4 System Description

The subsequent sections provide a list of the submodules used. We will describe why and how each model was utilized, and assess its relevance to our problem statement.

4.1 RAG using GEMINI

Large language models (LLMs) function best when Retrieval-Augmented Generation (RAG) Lewis et al., 2020 extends their capabilities to internal knowledge bases or specialized domains without requiring retraining. By guaranteeing that LLM output is accurate, pertinent, and usable in a variety of circumstances, this technique improves LLM output. Giving end users out-of-date or generic information when they're looking for specific answers is a prevalent problem with LLMs. This problem is solved by RAG, which instructs LLMs to obtain relevant information from reliable, pre-selected knowledge sources, improving accuracy and dependability.

Domain-specific or pertinent data is loaded, split into appropriately sized chunks to preserve context, and finally embedded using embedding models. The resultant embeddings are kept in a vector database so that documents with similar semantic content may be quickly retrieved. Data is then extracted from these embeddings according to how closely the query supplied by the user matches the documents. We use RAG with Gemini as the foundational LLM because of Gemini's extensive knowledge base as well as its large context window which allows chunks with higher semantic lengths to be supplied by the retriever.

LangChain simplifies the implementation of RAG by providing tools to load relevant datasets, such as the MedQA dataset, through its Data Loaders. It facilitates the chunking of data and the creation of embeddings using predefined functions and embedding models. The user's query is incorporated into a template and given as input into the LLM, while a Retriever component assists in finding similar documents based on query similarity. Utilizing MedQA data enhanced Gemini's answering ability, resulting in improved accuracy and relevance in responses. This integrated approach underscores the effectiveness of RAG in augmenting LLM performance specifically in the domain of Medical science.

4.2 Palmyra Quantised version

In our experiments, we employed a big decoder-only transformer model, known as Palmyra-20b. The Pile dataset Gao et al., 2020, which was tokenized with the GPT2 Radford et al., 2019 BPE tokenizer, served as the pre-training dataset for Palmyra-20b. It is a GPT-based model with 48 attention heads, a hidden size of 6144, 44 transformer layers, and a sequence length of 2048. The distributed Adam optimizer was used to train the model, which has two parallelism configura-

tions: pipeline parallelism of 1 and tensor parallelism of 4. Given the constraints of limited computational resources, we implemented 4-bit quantization on the model to mitigate computational demands while preserving efficiency. Quantization Gholami et al., 2021 is a technique that involves the process of converting the weights of the model from a higher precision to a lower precision. In our approach, we used 4-bit quantization to reduce the precision of weights and activations of Palmyra-20b to only 4-bit integer format. By quantization, we were able to significantly decrease memory and computational requirements without compromising model performance substantially to give accurate predictions by analyzing the symptoms provided to the model.

4.3 QLoRA on palmyra

Since fine-tuning LLMs like Palmyra20B is highly computationally expensive, we used PEFT (Parameter Efficient Fine Tuning) to make sure the training could be carried out on consumer-grade GPUs. In particular, we used QLORA Dettmers et al., 2023 (Quantized Low-Rank Adaptation) which quantizes a pre-trained model to 4-bit weights and adds an Adaptor - a low-rank tensor of trainable weights that can then be used to fine-tune the model through back-propagation. QLORA achieves far more efficient fine-tuning through the use of 4-bit Normal Float datatype which has been empirically proven to yield superior results to 4bit Floats. QLORA also employs double-quantization where not only are the weights but the quantization constants themselves are also quantized saving further memory. Finally, this approach uses Paged Optimisers allowing NVIDIA to manage memory effectively and ensuring that QLORA gives optimal results in parallel processing.

4.4 Proposed Model - Quantised Palmyra with RAG

In this, we incorporated 3 modules for the Error detection and correction task. The first one was the RAG module as explained in the previous section the second was the quantized and finetuned Palmyra med 20B and the third NER module.

We initially conducted Named Entity Recognition (NER) using GEMINI to identify words representing diseases or vaccines in the text. After masking these identified words, we implemented the Retrieval-Augmented Generation (RAG) model on an external dataset Jin et al., 2020. If the RAG score fell below a certain threshold, we passed the input to our model i.e. Palmyra(quantized and finetuned version). If the word provided by our Palmyra or RAG model matched the word detected by NER, no error was detected. Otherwise, if a different word is obtained from the model, then it is replaced with the masked word identified by NER. Finally, the error sentence is mapped with the sentence ID to get the output in the desired format. Our proposed model is illustrated in Figure 1, which provides a visual representation of the key components and relationships within our framework. For determining the error flag NER plays a pivotal role. It is so because irrespective of what flow the text takes (i.e. RAG or Palmyra), the output will always be compared with the NER's output for the error flag.

4.5 Metrics Used

To evaluate our model and assess its accuracy in light of the corrected sentence, we have adopted the following metrics for evaluation

4.5.1 R1-F

The ROUGE-1 F1-score is a metric commonly utilized in natural language processing tasks, particularly in the evaluation of automatic text summarization systems. ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation, focuses on measuring the quality of summaries generated by algorithms in comparison to human-generated reference summaries.

Specifically, the ROUGE-1 F1-score assesses the overlap of unigrams (individual words) between the generated summary and the reference summary. It is computed by taking into account both precision and recall of unigrams. Precision measures the proportion of correctly included unigrams in the generated summary relative to all unigrams present, while recall measures the proportion of correctly included unigrams relative to all unigrams in the reference summary.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{1}$$

Here, precision is the number of samples correctly predicted out of the number of samples predicted in that category. Recall is the number of samples predicted correctly out of the number of samples present for that class.

4.5.2 BERT SCORE

BERTScore is a collection of three metrics - BERT-Precision, BERT-Recall, and BERT-F1. As the names imply, BERT-Precision measures how well the candidate texts avoid introducing irrelevant content. BERT-Recall measures how well the candidate texts avoid omitting relevant content. BERT-F1 is a combination of both Precision and Recall to measure how well the candidate texts capture and retain relevant information from the reference texts.

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x_j} \in \hat{x}} \max_{x_i \in x} (x_i^T \cdot \hat{x}_j) \tag{2}$$

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} (x_i^T \cdot \hat{x}_j) \tag{3}$$

$$F1 = 2 \times \frac{P_{bert} \times R_{bert}}{P_{bert} + R_{bert}} \tag{4}$$

Model	Score			
	R1F Score	**BERT Score**	**BLEURT Score**	**Aggregate Score**
Quantised Palmyra	0.46277	0.48681	0.49753	0.482371
Quantised+QLoRa	0.54802	0.57079	0.55477	0.55786
Pure RAG	0.66376	0.64557	0.60720	0.63884
Quantised+QLoRa+RAG	0.**70306**	0.**74372**	0.**75217**	0.**73298**

Table 2: Scores for various model

4.5.3 BLEURT

BLEURT (Bilingual Evaluation Understudy with Representations from Transformers) Sellam et al., 2020 is a novel, machine learning-based automatic metric for Natural Language Generation BLEURT that can capture non-trivial semantic similarities between sentences. It takes a pair of sentences as input, a reference, and a candidate, and it returns a score that indicates to what extent the candidate is fluent and conveys the meaning of the reference.

4.5.4 Aggregate Score

The aggregate score is calculated as the Mean of ROUGE-1-F, BERTScore, and BLEURT

$$Aggregate = \frac{R1F + BERTScore + BLEURT}{3} \quad (5)$$

Parameter	**Value**
per_device_train_batch_size	4
gradient_accumulation_steps	4
optim	paged_adamw_32bit
logging_steps	1
learning_rate	1e-4
fp16	True
max_grad_norm	0.3
num_train_epochs	2
evaluation_strategy	steps
eval_steps	0.2
warmup_ratio	0.05
save_strategy	epoch
group_by_length	True
save_safetensors	True
lr_scheduler_type	Cosine
Seed	42

Table 3: Hyperparameters for Fine Tuning

5 Experimental Setup

We primarily used Google Colab notebooks for our workflow as well as for less computationally demanding tasks such as NER, EDA, text masking, RAG, etc.

Colab notebooks provide free access to a single T4 GPU (12GB RAM, 8GB VRAM, 64GB disk space). However, running quantized LLMs or fine-tuning had much higher computational requirements, and we therefore used Kaggle notebooks, which provide limited access to 2x T4 GPUs (15 GB of VRAM each). Please refer to Table 3 for a comprehensive overview of the parameters employed during the fine-tuning process. Since dataset preparation requires disk storage and frequent reads and writes, we use Jupyter Kernels for the same. We used the BitsAndBytes library for 4-bit quantization as well as the PEFT, Accelerate, and Datasets libraries by Huggingface for fine-tuning.

For performing NER on text, we used the GEMINI API from Google AI Studio. It had a maximum query limit of 60 queries per second. Since we were using GEMINI for NER as well as for RAG, this became our bottleneck, which sometimes led the session to crash. To address this, we imposed a timeout after every few API calls as well as made frequent local saves to the inferred results.

We implemented RAG using the Langchain framework, using GEMINI as our LLM. For implementing retrieval in our knowledge base, we used GEMINI embeddings to populate our vector store, which was a locally created ChromaDB instance.

6 Result

In our study, we employed a series of approaches aimed at enhancing the accuracy of our model. Initially, we implemented the quantized Palmyra approach, in which we tested the model that we built after the 4-bit quantization of Palmyra-20b. This gave a modest aggregate score of 0.482371. However, recognizing the room for improvement, we continued to refine our methodology. Building upon the quantized palmyra framework, we introduced the quantized+ QLoRa approach. In quantised palmyra, we fine-tuned using QLora on MEDQA data, which demonstrated a notable improvement, yielding an aggregate score of 0.55786. Encouraged by this progress, we further augmented our model with the Pure RAG technique, resulting in a substantial enhancement in aggregate score to 0.63884. Finally, through the integration of all three approaches—quantized, QLoRa, and RAG—into our model, we achieved the highest aggregate score of **0.73298**. The detailed scores for each approach are described in Table 2.

7 Limitations

The model struggles to give the correct output if the error is not related to a disease or pathogen. NER plays a crucial role in detecting pathogens or diseases from the text and therefore proves to be a bottleneck for accuracy since if NER fails to accurately determine the

disease, pathogen, or suggestion, the result will not be accurate regardless of the robustness of the model. The RAG approach fails for symptoms that are phrased very differently from those in the principal texts.

8 Future Work

Using Larger and More Powerful LLMs: Larger LLMs like Meditron-70b and Palmyra-med-40b can be used for achieving better accuracy in error detection and correction in clinical notes given sufficient computational power. The greater number of weights in these larger models allows them to capture more intricate patterns and nuances in the data during training.

FineTuning on a larger dataset, which will contain richer and more diverse medical information, can improve the model performance. Integrating multimodal information, such as images or structured data from electronic health records, alongside text data could provide richer context and improve error detection and correction accuracy.

Enhancing Model Robustness: The model can be made more robust against failures by having an end-to-end architecture where individual modules like NER, error detection, etc. are not carried out independently.

9 Conclusion

To conclude with this work for the MEDIQA-CORR task at NAACL, In ClinicalNLP 2024, we investigated four approaches for detecting and correcting errors in clinical notes. Our experiments demonstrated that the combined approach of Quantised Palmyra with RAG achieved the best performance, with an aggregate score of 0.73298. However, a key limitation identified is the reliance on named entity recognition (NER). Errors in NER can impact the overall performance of the system. Looking towards the future, research efforts should focus on mitigating the dependence on NER. Additionally, exploring alternative techniques and leveraging a larger, more comprehensive dataset holds promise for further improving the accuracy of error detection and correction in clinical notes. This will ultimately lead to a more robust and reliable system for enhancing the quality of clinical documentation.

References

Asma Ben Abacha, Wen wai Yim, Velvin Fu, Zhaoyi Sun, Fei Xia, and Meliha Yetisgen. 2024. Overview of the mediqa-corr 2024 shared task on medical error detection and correction. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Daniil A. Boiko, Robert MacKnight, and Gabe Gomes. 2023. Emergent autonomous scientific research capabilities of large language models.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. Meditron-70b: Scaling medical pretraining for large language models.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Ran Elgedawy, Sudarshan Srinivasan, and Ioana Danciu. 2024. Dynamic qa of clinical documents with large language models.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams.

Mingyu Jin, Qinkai Yu, Dong Shu, Chong Zhang, Lizhou Fan, Wenyue Hua, Suiyuan Zhu, Yanda Meng, Zhenting Wang, Mengnan Du, and Yongfeng Zhang. 2024. Health-llm: Personalized retrieval-augmented disease prediction system.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

David Oniani, Jordan Hilsman, Hang Dong, Fengyi Gao, Shiven Verma, and Yanshan Wang. 2024. Large language models vote: Prompting for rare disease identification.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-lm: Training multi-billion parameter language models using model parallelism.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. 2021. Understanding the capabilities, limitations, and societal impact of large language models.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine.

Dong Yuan, Eti Rastogi, Gautam Naik, Sree Prasanna Rajagopal, Sagar Goyal, Fen Zhao, Bharath Chintagunta, and Jeff Ward. 2024. A continued pretrained llm approach for automatic medical note generation.

Yinghao Zhu, Changyu Ren, Shiyun Xie, Shukai Liu, Hangyuan Ji, Zixiang Wang, Tao Sun, Long He, Zhoujun Li, Xi Zhu, and Chengwei Pan. 2024. Realm: Rag-driven enhancement of multimodal electronic health records analysis via large language models.