Leveraging AI for Automatic Generation of Medical Notes and Clinical Assistance

By William Zheng

Abstract:

The importance and prevalence of Large Language Models (LLMs) has been increasing year by year. Whether it be in medicine, literature, datascience, or entertainment, the potential to apply LLMs to many fields has only grown. In this paper, we seek to apply LLMs in the medical field to specifically aid in clinical notes generation. Doctors and physicians dedicate a substantial amount of their time to writing clinical notes for their patients [2]. Therefore, a means to help automatically generate clinical notes would help save the time of doctors and physicians. The goal of this project was to leverage LLMs to not only help improve the creation of clinical notes, but to also automate some simple tasks. In this paper, we tested a new framework for generating clinical notes, and found that zero-shot prompting performs the best.

Introduction:

MemGPT is a LLM framework that can leverage an extensive temporal context by simulating enhanced memory [3]. This memory is organized like an operating system, differentiating between working memory and long-term storage. MemGPT achieves this by utilizing callable tools that LLM agents can directly call to perform specific memory-related tasks, such as retrieving information from archival memory. In this framework, the LLM decides what to store in archival memory and what to keep in its current context [3]. This method allows LLMs to go beyond restraints due to contextual memory size.

The Medinote project aimed to develop a model specialized in creating clinical notes. In addition to releasing a fine tuned model, the team also provided the framework for fine tuning the model for clinical note generation [1]. Medinote aimed to generate clinical notes from patient/doctor conversations, but genuine conversations are illegal to record under U.S. regulation. To get around this, Medinote used NoteChat to generate synthetic patient/doctor conversations [1]. NoteChat utilized the PMC dataset, which consisted of clinical case studies, to generate the patient doctor conversations [4].

System:

To make clinical note generation friendly to use, we have implemented a tool using the MemGPT's framework that allows a LLM agent to call another model specifically for clinical note generation (under the Medinote framework). MemGPT operates by using a hierarchical memory system inspired by operating systems to manage the limited context windows of LLMs. It divides memory into a main context (containing system instructions, a writable working context for critical facts, and a rolling FIFO queue for recent interactions) and external context (recall storage for evicted data and archival storage for large datasets). A Queue Manager handles memory overflow by moving data to external storage. The Function Executor enables MemGPT to perform tasks like retrieving, modifying, and reinserting data into the main context using function calls. In our use case, physicians can upload documents to the MemGPT framework to chat with the LLM agent. Documents could be patient histories, test results, or clinical notes for example. Furthermore, MemGPT's framework allows for the future integration of additional tools that the LLM agent can leverage to assist physicians.

The researchers of Medinote have released a llama based model for clinical note generation in addition to providing the framework for their finetuning and evaluation methods. However, Medinote's framework can be split into two parts, direct clinical note generation and in-direct clinical note generation [1]. Direct clinical note generation refers to generating clinical notes directly from patient doctor conversations. In-direct clinical note generation refers to generating clinical notes from patient summaries instead [1]. These patient summaries are produced by an LLM extracting patient information from patient-doctor conversations, and filling that information into a provided template. The researchers found that the in-direct framework performed worse than the direct method [1]. Therefore, in our system, we use the direct method as well. Figure 1 below displays the whole system.

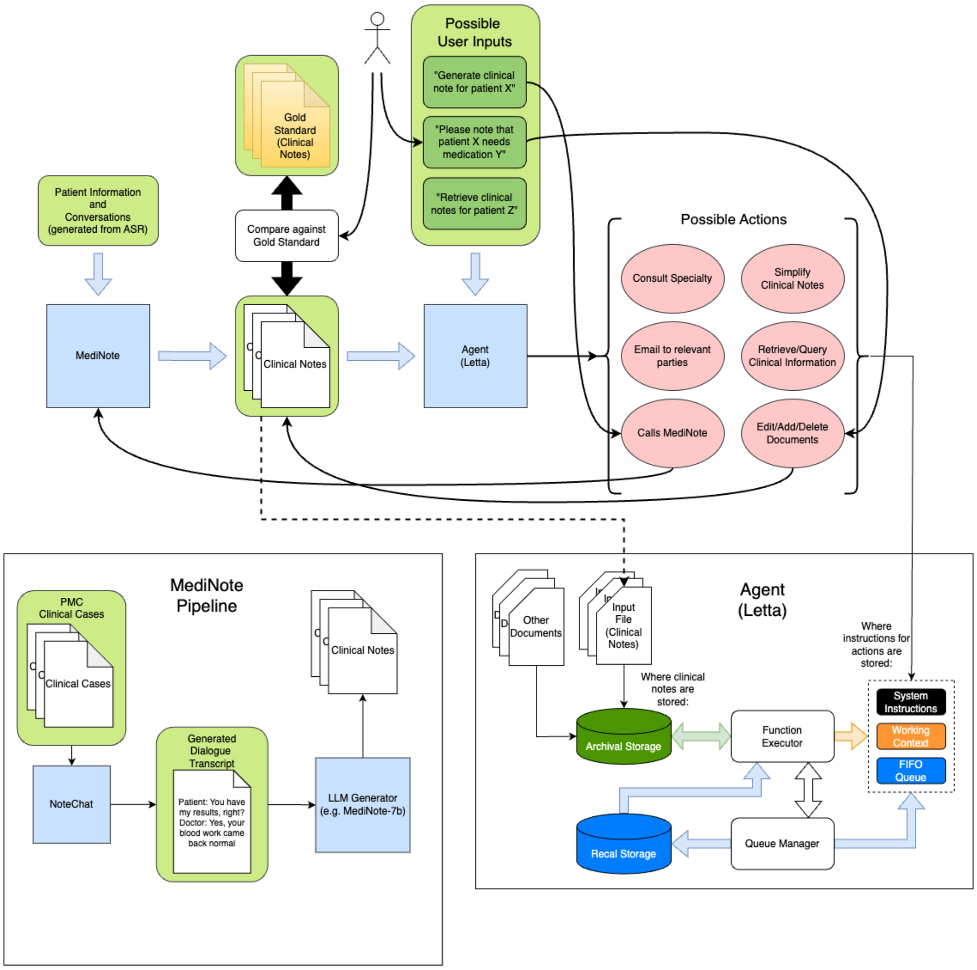

Possible User Inputs
"Generate clinical note for patient X"
"Please note that patient X needs medication Y"
"Retrieve clinical notes for patient Z"
Gold Standard (Clinical Notes)
Compare against Gold Standard
Patient Information and Conversations (generated from ASR)
MediNote
Clinical Notes
Agent (Letta)
Possible Actions
Consult Specialty
Simplify Clinical Notes
Email to relevant parties
Retrieve/Query Clinical Information
Calls MediNote
Edit/Add/Delete Documents
MediNote Pipeline
PMC Clinical Cases
Clinical Cases
NoteChat
Generated Dialogue Transcript
Patient: You have my results, right? Doctor: Yes, your blood work came back normal
LLM Generator (e.g. MediNote-7b)
Clinical Notes
Agent (Letta)
Other Documents
Input File (Clinical Notes)
Where clinical notes are stored:
Where instructions for actions are stored:
System Instructions
Working Context
FIFO Queue
Archival Storage
Function Executor
Recal Storage
Queue Manager

Figure 1: Here is the outline of the whole system. Users are able to first interact with MemGPT to make a call to Medinote for clinical note generation. Afterwards, users are able to make queries pertaining to the clinical note. The MediNote pipeline shows the direct clinical note generation method, which takes conversations as input directly.

Methods:

To improve upon clinical note generation, we investigated if existing real clinical notes can be used directly to improve clinical note generation. In particular, we tested if one-shot prompting would improve clinical note generation. In Medinote's Jupiter tutorial, researchers noted briefly to have investigated into one shot prompting. However, they ultimately decided to not pursue this direction and did not elaborate on why they did not choose this direction. In their investigation, they implemented K-Shot learning by including both the patient conversation and the corresponding real clinical note. However, these few-shot examples rarely match the content of the given patient doctor conversations.

Therefore, we investigated if choosing a specific real clinical note, that is similar in topic with the conversation, will improve the performance of clinical note generation. This is done by first converting each real clinical note into a short summary of the patient's symptoms and treatments, which is done by GPT 3.5 Turbo. The summaries are then turned into sentence embeddings through a sentence transformer. This summarization step is used since converting full clinical notes into embeddings to then look up can be computationally expensive. Through this, we created a corpus of embeddings that match to a specific clinical note. When we want to generate a new clinical note for a conversation, we run the same steps. The conversation is first summarized by GPT 3.5 Turbo, and then converted into embeddings. We can then do a cosine similarity search between our conversational embeddings and the corpus of embeddings to find the highest matching clinical note to the conversation. The most similar clinical note is then passed through the prompt to the LLM mode for clinical note generation. The full process is shown in Figure 2 below.

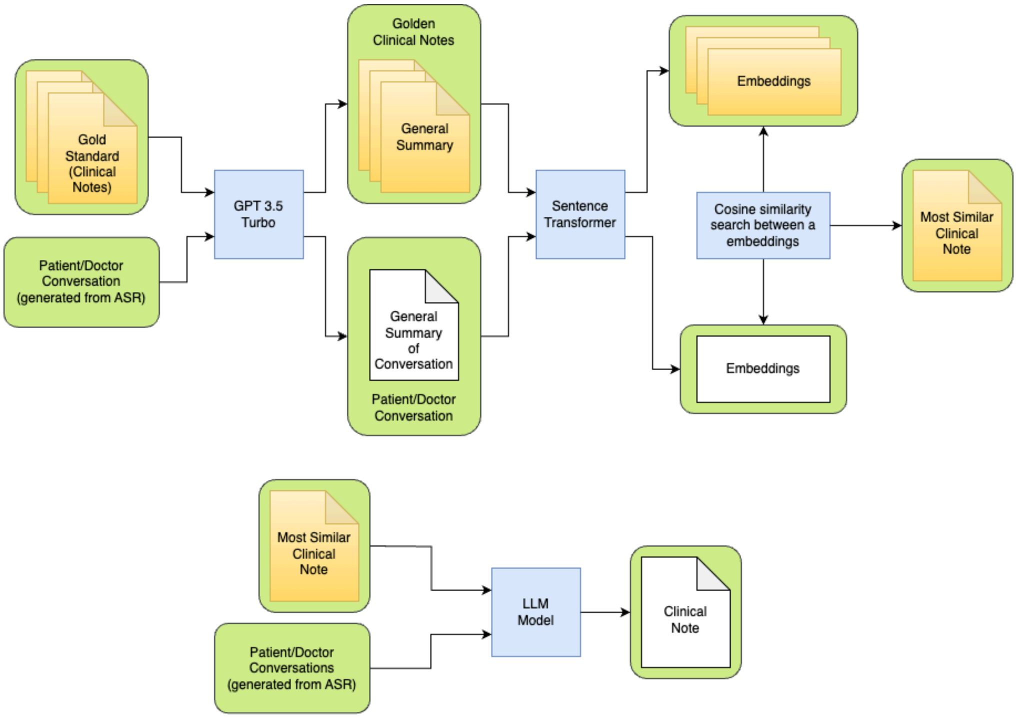

Gold Standard (Clinical Notes)
Patient/Doctor Conversation (generated from ASR)
GPT 3.5 Turbo
Golden Clinical Notes
General Summary
General Summary of Conversation
Patient/Doctor Conversation
Sentence Transformer
Embeddings
Cosine similarity search between a embeddings
Embeddings
Most Similar Clinical Note
Most Similar Clinical Note
Patient/Doctor Conversations (generated from ASR)
LLM Model
Clinical Note

Figure 2:
Given a patient-doctor conversation, the most related clinical note to the conversation is found and used in the generation of the clinical note of the conversation.

Results:

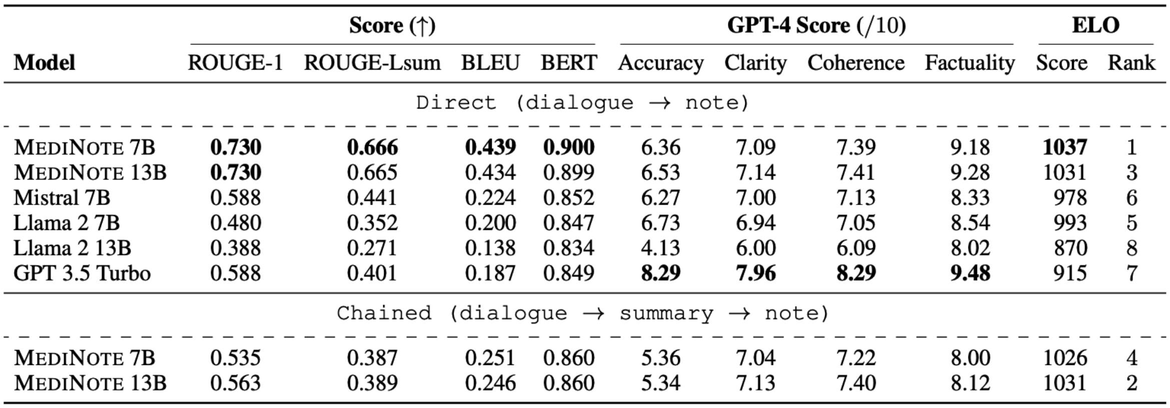
In the original Medinote paper, they evaluated their fine tuned Medinote models against the GPT 3.5 Turbo models under the direct generation framework. However, as previously stated, the in-direct method of generating clinical notes from patient summaries proved ineffective. Therefore, the researchers did not test the rest of the models under the in-direct framework. To understand why and see if GPT 3.5 Turbo would benefit from the in-direct method, we replicated the in-direct generation method. From a few inferences from the in-direct method, we found issues contributing to the poor performance. The biggest problem was the fact that the additional strict step of creating patient summaries in a rigid template often caused information from the conversation to be lost. Additionally, having a strict template for patient summaries relies heavily on the LLM model getting the syntax right. It was for these reasons that we also disregarded the in-direct method of generating patient summaries from conversations.

For our experiments, we first tested the performance of GPT 3.5 Turbo with one-shot examples at different sample sizes (686 vs 2120 samples). GPT 3.5 Turbo one-shot was given an example of a clinical note similar in topic to the conversation input, as described in the Methods section. From this test, we found that there was no significant difference in performance of GPT 3.5 Turbo with one-shot examples when sample size changed. However, it should be noted the same cannot be said for GPT 3.5 Turbo zero-shot, with direct clinical note generation, as tested in the original Medinote paper and listed in Table 1. GPT 3.5 Turbo under the same framework as the Medinote paper, performed worse under a smaller sample size.

Additionally, we wanted to know how significant Medinotes's performance would change if evaluated on a smaller test set. Medinote 7B was then tested on a sample size of 200 under the direct generation framework, with results shown in Table 2. The change in performance on the metrics from evaluating on a smaller dataset was no larger than ± 0.02 points, in comparison to the Medinote 7B model evaluated also under the direct framework on the full dataset (as listed in Table 1). By evaluating on a smaller dataset, we were able to perform inference and evaluation much quicker with less computation resources. From our testing, we found that using a smaller dataset still allowed us to preserve accuracy and information for Medinote's and GPT 3.5 Turbo's one-shot performance.

From our evaluation, we have found that GPT 3.5 Turbo, with specifically chosen one-shot examples, does not significantly improve or decrease the performance when compared to GPT 3.5 Turbo with no examples. In particular, GPT 3.5 Turbo with no-shot examples performed only marginally worse, with being only at most 0.058 points behind. In either case, both methodologies were out performed by Medinote with more than a 0.5 point lead across all metrics tested, as seen in Table 2.

Table 1: This table is from the **original** Medinote paper, which records the evaluation of clinical note generation from the direct and in-direct methods [1].



	Score (↑)				**GPT-4 Score** (/10)				**ELO**	
Model	ROUGE-1	ROUGE-Lsum	BLEU	BERT	Accuracy	Clarity	Coherence	Factuality	Score	Rank
`Direct (dialogue → note)`										
MEDINOTE 7B	**0.730**	**0.666**	**0.439**	**0.900**	6.36	7.09	7.39	9.18	**1037**	1
MEDINOTE 13B	**0.730**	0.665	0.434	0.899	6.53	7.14	7.41	9.28	1031	3
Mistral 7B	0.588	0.441	0.224	0.852	6.27	7.00	7.13	8.33	978	6
Llama 2 7B	0.480	0.352	0.200	0.847	6.73	6.94	7.05	8.54	993	5
Llama 2 13B	0.388	0.271	0.138	0.834	4.13	6.00	6.09	8.02	870	8
GPT 3.5 Turbo	0.588	0.401	0.187	0.849	**8.29**	**7.96**	**8.29**	**9.48**	915	7
`Chained (dialogue → summary → note)`										
MEDINOTE 7B	0.535	0.387	0.251	0.860	5.36	7.04	7.22	8.00	1026	4
MEDINOTE 13B	0.563	0.389	0.246	0.860	5.34	7.13	7.40	8.12	1031	2

Table 2: This table records the evaluation of our system based on the same metrics used in the original paper. Specifically, we evaluated the GPT 3.5 Turbo one-shot example; its implementation is shown in Figure 2. We evaluated models with a smaller sample size in comparison to the Medinote paper.

Model and setup	Num Samples	Rouge 1	Rouge 2	Rouge L	Rouge Lsum	Bleu	Bert
GPT 3.5 Turbo, with one-shot	686	0.479	**0.300**	**0.401**	**0.401**	**0.197**	0.801
	2120	0.479	0.303	0.404	0.404	0.200	0.800
GPT 3.5 Turbo, directly from conversation	686	**0.491**	0.275	0.390	0.390	0.139	**0.854**
MediNote 7B. directly from conversations	200	0.736	0.585	0.681	0.681	0.459	0.91

Similarity Metrics:

For our evaluation, we looked into a few metrics that measure the similarity between two texts. In particular, we used the ROUGE metric that evaluated the generated clinical notes against a real clinical note, based on overlapping units such as n-grams, word sequences, or sentence-level structures. ROUGE-1: evaluates by single word overlap, while ROUGE-2 evaluates by two consecutive words overlap. ROUGE-L focuses on the longest sequence of words that appear in both the candidate and reference texts while maintaining their order. This metric captures fluency and grammatical quality.

ROUGE-Lsum is a tailored version of ROUGE-L for document-level summarization, measuring overlaps at the sentence level. Bleu assesses how well the generated text matches a reference text, using a precision-based approach that considers the overlap of words and phrases (n-grams) between the two. Lastly, we also evaluated our system on the BERT metric. BERT leverages semantic similarity by using contextual embeddings from BERT (Bidirectional Encoder Representations from Transformers) to determine similarity.

Conclusion:

This study explored and assessed various methodologies to enhance the generation of clinical notes. Building upon existing tools and creating new ones, we aimed to contribute to the development of solutions that could ultimately support clinicians and physicians. Our findings indicate that the most effective approach for generating clinical notes is direct generation from patient-doctor conversations, leveraging the fine-tuned capabilities of the Medinote models.

Future Work:

Potential future work on this project includes testing on larger and more diverse datasets, implementing more clinical note generation improvements, and other tools for MemGPT. We would also like to look into other methods of fine tuning LLMs, whether that be Medinote, Llama, or etc, into better clinical note generation.

Acknowledgments:

I would like to take the time to say that I am incredibly grateful for Filippos's guidance and support throughout this semester, as well as for Dr. Corso's generosity in providing me the opportunity to work on this project. Through this project, we feel that we gained valuable experience that will allow me to pursue this field further.

Citation:

[1] EPFL-IC-Make-Team, “medinote/report.pdf at main · EPFL-IC-Make-Team/medinote,” *GitHub*, 2023. Available: https://github.com/EPFL-IC-Make-Team/medinote/blob/main/report.pdf. [Accessed: Dec. 18, 2024]

[2] E. Joukes, A. Abu-Hanna, R. Cornet, and N. de Keizer, “Time Spent on Dedicated Patient Care and Documentation Tasks Before and After the Introduction of a Structured and Standardized Electronic Health Record,” *Applied Clinical Informatics*, vol. 09, no. 01, pp. 046–053, Jan. 2018, doi: https://doi.org/10.1055/s-0037-1615747

[3] C. Packer, V. Fang, S. G. Patil, K. Lin, S. Wooders, and J. E. Gonzalez, “MemGPT: Towards LLMs as Operating Systems,” *arXiv.org*, Oct. 12, 2023. doi: https://doi.org/10.48550/arXiv.2310.08560. Available: https://arxiv.org/abs/2310.08560

[4] J. Wang *et al.*, “NoteChat: A Dataset of Synthetic Patient-Physician Conversations Conditioned on Clinical Notes,” *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 15183–15201, Jan. 2024, doi: https://doi.org/10.18653/v1/2024.findings-acl.901

Appendix:

Resources:
GitHub Repo: https://github.com/WilliamUMICH/LettiNote/tree/master

Meeting Notes:
https://docs.google.com/document/d/1yahWqg7lCa-4-vi5NOFVNmFB-GxZz8CVFww5vvUnTOg/edit?usp=sharing