SpecialtyScribe: Enhancing SOAP note Scribing for Medical Specialties using LLM's

Sagar Goyal[*]
DeepScribe Inc.
sagar@deepscribe.tech

Eti Rastogi[*†]
DeepScribe Inc.
eti_rastogi@yahoo.com

Fen Zhao
DeepScribe Inc.
fen@deepscribe.tech

Dong Yuan[†]
DeepScribe Inc.
doffery20@gmail.com

Andrew Beinstein
DeepScribe Inc.
andrew.beinstein@deepscribe.tech

Abstract

The healthcare industry has accumulated vast amounts of clinical data, much of which has traditionally been unstructured, including medical records, clinical data, patient communications, and visit notes. Clinician-patient conversations form a crucial part of medical records, with the resulting medical note serving as the ground truth for future interactions and treatment plans. Generating concise and accurate clinical SOAP (Vivek Podder, 2022) notes is critical for quality patient care and is especially challenging in specialty care, where relevance, clarity, and adherence to clinician preferences are paramount. These requirements make general-purpose LLMs unsuitable for producing high-quality specialty notes. While recent LLMs like GPT-4 and Sonnet 3.5 have shown promise, their high cost, size, latency, and privacy issues remain barriers for many healthcare providers.

We introduce SpecialtyScribe, a modular pipeline for generating specialty-specific medical notes. It features three components: an Information Extractor to capture relevant data, a Context Retriever to verify and augment content from transcripts, and a Note Writer to produce high quality notes. Our framework and in-house models outperform similarly sized open-source models by over 12% on ROUGE metrics. Additionally, these models match top closed-source LLMs' performance while being under 1% of their size. We specifically evaluate our framework for oncology, with the potential for adaptation to other specialties.

1 Introduction

The healthcare industry relies on storing, processing, and referencing large amounts of clinical and research data, such as patient records, conversations, treatment histories, and medical research. Most of this data is unstructured and language-based, making it challenging to extract relevant information. Traditional NLP methods, and more recently Large Language Models (LLMs), have enabled efficient analysis to improve diagnoses, personalized treatments, and health outcomes. With increasing digitization, medical records are now maintained electronically as electronic health records (EHRs), with tools to add structure to notes. A medical visit note, the doctor's concise summary of medically relevant information, is critical for long-term reference and guiding future interactions.

Generating accurate medical notes from clinician-patient conversations is crucial for high-quality care. These notes reduce the administrative burden, enhance record accuracy, and ensure information is accessible for decision-making (Berg, 2023). However, generating high-quality notes in specialized fields like oncology is challenging due to high requirements for relevance, brevity, specificity, and clarity. Before LLMs, models like T5 or BART fine-tuned for note generation faced issues like nonfactual content (Chelli et al., 2024). Although newer LLMs (e.g., Opus, Sonnet, GPT-4) have potential, they are costly and pose privacy concerns for many healthcare facilities. Fine-tuning public LLMs (Goyal et al., 2024; Yuan et al., 2024) has been explored to improve general medical note generation.

A significant challenge in using generative models like LLMs is hallucination: "generated content that is nonsensical or unfaithful to the provided source content" (Ji et al., 2023). Inaccurate information in medical notes can severely impact quality and reliability. Oncology requires specific and concise note-taking focused on primary cancer diagnoses. Colorectal surgeons, for example, prioritize cancer-related treatments, with general symptoms included only if relevant to the treatment plan. Thus, oncology notes must be selective,

[*]These authors contributed equally to this work.
[†]Work done while at DeepScribe.

Proceedings of the Second Workshop on Patient-Oriented Language Processing (CL4Health), pages 34–45

emphasizing critical information to support cancer care.

We address these challenges by focusing on key aspects of oncology note generation:

- *Completeness*: covering all essential information
- *Conciseness*: avoiding irrelevant details
- *Writing Quality*: ensuring readability, clarity and medical language flow
- *Organization*: categorizing information correctly in the SOAP note

Our approach simplifies note creation through three key modules. The Information Extraction module captures oncology-specific details. The Context Retriever gathers additional context, verifies accuracy, and reduces hallucinations. The Summarizer generates a medical note, ensuring precision and reliability.

Our contributions include:

- A unique three-step approach with an Information Extractor, Context Retriever, and Summarizer to generate high-quality specialty notes.
- Fine-tuned LLM-based models to extract key medical concepts and also write the final note. These models outperform similar sized open-source models by more than 100% and match closed source models while being less than 1% the size of them
- An embedding-based verification and augmentation method to minimize hallucinations and improve recall.
- Demonstration of our framework's effectiveness in clinical settings, matching the performance of top LLMs.

2 Related Work

Medical Note Generation. Generating high-quality medical notes from doctor-patient conversations is a challenging task. Prior to the advent of large language models (LLMs), previous approaches attempted to address this problem by breaking it into multiple stages (Krishna et al., 2020)—first identifying key transcription snippets, grouping them, and then summarizing—or by chunking the transcription (Zhang et al., 2021) into smaller pieces. However, these models failed to achieve real-world usable quality.

With the emergence of LLMs, recent works (Van Veen et al., 2023; Biswas and Talukdar, 2024; Goyal et al., 2024) have focused on leveraging or prompting powerful private LLMs, such as GPT-4 and MedPaLM, to enhance medical note generation. These models have a better understanding of language and can produce more readable text. However, reliance on private vendors raises concerns about data privacy and incurs high costs.

This has driven further research (Yuan et al., 2024; Kerner, 2024) into developing specialized medical LLMs that are better equipped to understand clinical texts and generate quality notes for general scenarios. Nonetheless, in oncology, the focus of medical note generation differs, and none of the existing approaches can be directly applied to oncology data without significant adaptation.

Information Extraction. To extract information from transcription text data, Named Entity Recognition (NER) or similar sequence tagging methods are often used to identify and extract key entities and information. Models such as BioBERT (Lee et al., 2020), MedBERT (Rasmy et al., 2021), and ClinicalBERT (Huang et al., 2019) have proven effective in this context. When combined with techniques for extracting entity relationships (Lv et al., 2016), events, or temporal information (Styler IV et al., 2014), these models can provide a comprehensive understanding of medical information from transcriptions. Recently, the use of large language models (LLMs) like MedPaLM (Singhal et al., 2023), PMC-LLaMA (Wu et al., 2024), or MEDITRON (Chen et al., 2023b) has made it more feasible to extract key information from transcriptions through prompting. However, these LLMs are still limited by their capabilities and may not always capture information accurately and comprehensively.

Summarization. Existing summarization approaches often focus on general abstractive summarization (Gupta and Gupta, 2019; Basyal and Sanghvi, 2023), or domain-specific tasks like news summarization (Zhang et al., 2024). However, generating medical notes requires more than just summarization; it demands attention to medical details and selective extraction of key information specific to different specialties.

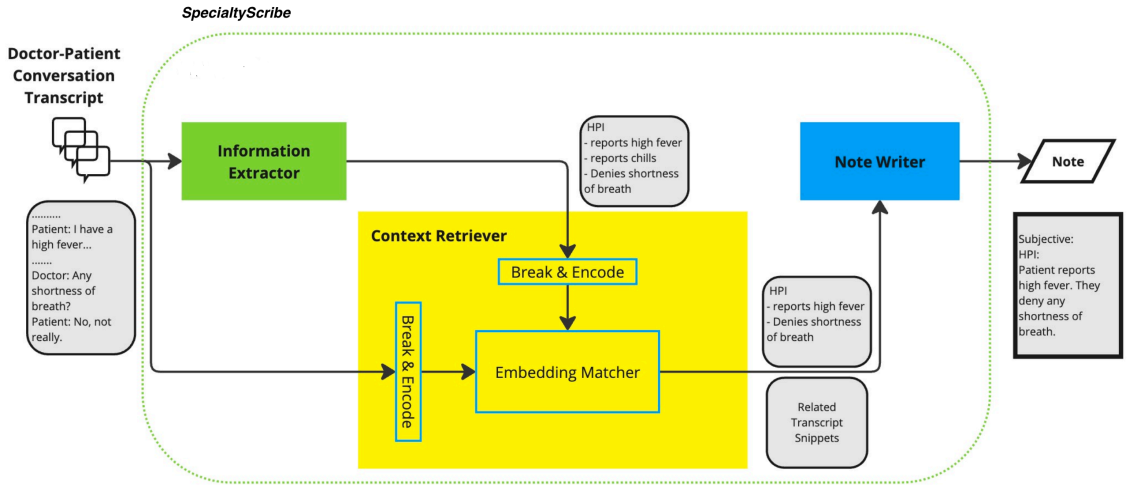


Figure 1: SpecialtyScribe Framework for the HPI section of a medical note from a doctor-patient conversation transcript

3 SpecialtyScribe

SpecialtyScribe consists of three primary modules: Information Extractor, Context Retriever, and Note Writer. Figure 1 illustrates the end-to-end functioning of the SpecialtyScribe framework using a basic example.

Information Extractor Module: This module takes the transcription as input and extracts specialty-specific (oncology) medically relevant information.

Context Retriever Module: This module generates additional transcript context to augment the extracted information and mitigates hallucinations by verifying the extracted information against the transcript. It takes the original transcript and the output of the Information Extractor Module as input. Transcript snippets are selected by splitting the transcript into sentence chunks and comparing the embeddings of the extracted information with those of the snippets, and selecting the top-k snippets to enhance the Note Writer model's context. We also use a hallucination detection algorithm to further filter the extracted information

Note Writer Module: This module generates the final medical note using the outputs of the Context Retriever Module, the extracted information (now filtered) and relevant transcript snippets. Since each section of SOAP note can have multiple subsections, (e.g. HPI, Chief Complaint, Medications etc.). This model is trained to generate subsection notes that combine to create the final note. It can also ignore irrelevant information that is part of the context.

3.1 Information Extractor

Our challenge involved working with a single, long transcript. Although newer LLMs can process longer texts (up to 32k tokens or more), they still face issues such as significant performance degradation depending on the relevant position of the information in the prompt, as discussed in (Liu et al., 2023). Traditional segmentation methods failed, as the model lacked full context and produced contradictory results. Additionally, we required a prompt-based extraction system capable of adapting to new

instructions to support customization requests by doctors. To address these issues, we reformulated information extraction as an Orca-style instruction task (Mukherjee et al., 2023). Here, the model's objective was to follow specific rules and extract information from given snippets. This approach was developed based on (Yuan et al., 2024), which describes the creation of a medical LLM that understands the nuances of spoken medical language and the structure of medical notes.

Training Data Generation: We began by breaking oncology notes and categorizing information into sub-sections, such as Cancer Procedures, Cancer Tests, Cancer Symptoms, and Current Symptoms. For each sub-section, we crafted specific instructions. See Appendix-B for more details.

Protecting Data and Controlling Costs: We robustly de-identified any PHI(Protexted Health Information) and PII (Personally Identifiable Information) as defined by HIPAA and US government respectively in the transcripts and notes by adapting the Microsoft Presidio library for our specific use case. This is discussed in more detail in Section 6. We incurred a one-time cost for preparing our training data by using GPT-4-32k. However, this cost was minimal compared to what would be required to serve these models in production at scale. We used GPT-4-32k to process 7,000 doctor-patient conversations, each ranging from 5 to 60 minutes with an average duration of 20 minutes, to create the OncNoteGen Dataset. This resulted in approximately 68,000 samples with an average context length of 7,000 tokens. To mitigate overfitting in information extraction tasks, we used two stages of tuning. First, we warmed the model with general instructions, including around 100,000 examples sampled from MedMCQA (Pal et al., 2022), PubMedQA (Jin et al., 2019), and general instruction datasets such as Orca (Mitra et al., 2023) and MetaMath (Yu et al., 2024). Second, we trained the model with our proprietary 68,000-sample oncology note data—OncNoteGen.

Following initial fine-tuning, we observed that the model struggled to distinguish between past, present, and future tenses, especially when identifying medications and doctor's orders. This issue appeared to be inherited from the GPT-4-32k model used to build the training dataset. To address this, we introduced an additional 3,000-4,000 QA-based instructions specifically designed to help the model understand these tense distinctions. An example prompt for this task is provided in Appendix-C.

3.2 Context Retriever

We developed an algorithm to identify the context from the transcript for the content generated by the information extractor. We decomposed the extracted information into pieces (e.g. by bullets generated from the extractor), and then used their embeddings to encode each piece of information. Similarly, we indexed the transcript, by chunking it into groups of varied sentence counts e.g. 1, 2, 5 and calculating their embeddings. Then we used embedding matching to find the transcript context for each piece of extracted information. We utilized the *all-mpnet-base-v2* model (Reimers and Gurevych, 2019) for generating embeddings and employed the *similarity_search_with_relevance_scores* function from Meta's FAISS library (Douze et al., 2024) to conduct embedding similarity searches. As the transcripts are divided into chunks by varying sentence numbers, it's possible to have duplicate sentences in the matched snippets. To address this, we removed duplicate sentences and arranged the sentences in the snippets in their original chronological order.

Hallucination Mitigation: In our framework, hallucinations can originate from two major sources. First, the Information Extractor can output some data which has no grounding in the transcript or the prompt and second, the example used in the few-shot prompt can propagate into or influence the output. To address the first kind, the Context Retriever first filters out the extracted content that does not have any transcript context support retrieved as explained in Algorithm-1 (see Appendix A for step by step explanation)

3.3 Note Writer

Final Note Generation: We trained the Note Writer model to generate notes based on the filtered extracted content and the corresponding contextual transcript. This model was trained on a diverse set of 1,000 human-expert-annotated notes. The experts annotated the data in two stages: first, they identified the relevant transcript snippets for each note subcategory; then, they combined these snippets to create a medically accurate subsection of the note. Since, each note was divided into its constituent subsections (e.g., Subjective: Labs, Plan: Follow-Ups), we end up with an average of 10,000 data points in the training set. We deliberately train it on a diverse medical note dataset rather than oncology specific dataset as we intend

to use this model across multiple specialties. While it is possible to train the information extractor to also do the note writing to reduce inference burden in real-world applications, we found that with the proposed framework, training them separately provided better performance and greater flexibility for use in other specialties.

We also developed a basic prompt that instructs the model to produce the note for each corresponding subsection. During training, the model learnt to create subsections of a note based on the retrieved relevant data, which were eventually combined into a complete note. This approach significantly reduced our context length requirements. The model was trained in a LoRA (Low-Rank Adaptation) setting, which made the training process fast, cost-effective, and scalable, with minimal impact on performance.

Algorithm 1 Information Filter

Input:
$I = \{i_1, i_2, \ldots, i_n\}$: Retrieved information set
T: Transcript
θ: Lower Bound Confidence
α: Similarity Confidence
E_p: Embeddings for examples from prompt
$E_T = ExtractEmbeddings(T)$
Output: $I_{included}$

Initialize included information $I_{included} = []$
for all information $i \in I$ **do**
  if i *in* T **then**
    $I_{included}.append(i)$
  else
    $E_i = ExtractEmbeddings(i)$
    Score = $EmbedMatch(E_i, E_T)$
    if Score $\geq \theta$ **then**
      $I_{included}.append(i)$
    end if
  end if
end for

for all $i_{incl} \in I_{included}$ **do**
  $E_i = ExtractEmbeddings(i_{incl})$
  PromptScore = $EmbedMatch(E_i, E_p)$
  TranscriptScore = $EmbedMatch(E_i, E_T)$
  if $PromptScore \geq \alpha \geq TranscriptScore$ **then**
    $I_{included}.remove(i_{incl})$
  end if
end for
return $I_{included}$

4 Experiment

4.1 Setup

Information Extraction: Consistent with the methodology described in (Yuan et al., 2024), our training utilized the pretrained version of Mistral-7B model. The learning rate was set at 2e-5 with cosine decay to 1e-5, and batch sizes were maintained at 128. Positional interpolation, referenced in (Chen et al., 2023a), addressed long-context management. Training occurred over 11 hours on 32 NVIDIA A100 GPUs distributed across four machines (8 GPUs per machine). Training perplexity and validation Rouge F1 scores for the OncNoteGen Dataset are shown in Figures 2, and 3 respectively.

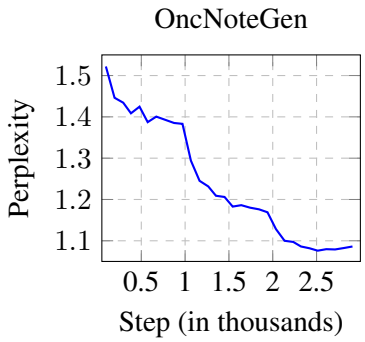
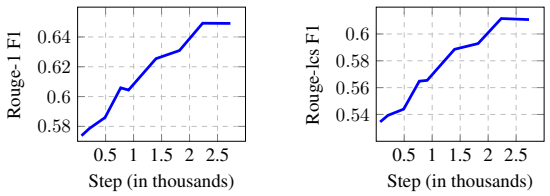


Figure 2: Training perplexity on OncNoteGen Dataset



(a) Rouge F1 (b) Rouge-lcs F1

Figure 3: Validation Rouge-1 F1 and Rouge-lcs F1 scores on OncNoteGen Dataset

Note Writer: We again utilized the pretrained version of Mistral-7B model described in (Yuan et al., 2024), as our base model. The model underwent training for two epochs with a batch size of 8. To enhance memory and cost efficiency during this process, we adjusted the Low-Rank Adaptation (Lora) rank to 32. Our computational resources included 8 NVIDIA RTX A6000 GPUs, each equipped with 48GB of memory, allowing for substantial parallel processing and data handling capabilities. During training sessions, the average GPU utilization was maintained at 85%, indicating efficient usage of hardware resources. Additionally, we integrated the FlashAttention 2 mechanism and utilized the DeepSpeed Zero 3 optimization framework to streamline our training process. The learning rate was set at 2e-5 with cosine decay to 1e-5.

Model	Missed	Redundant	Misclassified
Opus	0.37	0.11	0.10
Sonnet-3.5	**0.31**	**0.08**	0.05
GPT-4-32k	0.40	**0.08**	0.05
mistralai/Mistral-7B-Instruct-v0.2	0.46	0.18	0.10
meta-llama/Meta-Llama-3-8B-Instruct	0.45	0.28	0.06
BioMistral/BioMistral-7B	0.53	0.51	**0.03**
SpecialtyScribe (ours)	**0.37**	**0.08**	**0.05**

Table 1: Results on Oncology Entity Identification Task indicating average Missed, Redundant, and Misclassified entities (lower is better)

Model	Aci-bench (subTask B)		OncNoteGen		
	ROUGE_L	BLEU	ROUGE_L	BLEU	Human(4)
Opus	0.21	0.09	0.27	0.15	2.44
Sonnet-3.5	0.21	0.10	0.26	0.14	2.78
GPT-4o	0.20	0.09	0.29	0.17	2.95
mistralai/Mistral-7B-Instruct-v0.2	0.13	0.05	0.19	0.10	2.69
meta-llama/Meta-Llama-3-8B-Instruct	0.19	0.09	0.25	0.15	2.53
SpecialtyScribe (Note Writer)	**0.24**	**0.12**	**0.31**	**0.21**	**3.14**

Table 2: Results on Note Writing Quality Task (higher is better)

4.2 Evaluation

We performed a comprehensive evaluation of leading open-source and proprietary models to assess the effectiveness of our Information Extraction (IE) model as well as the note-generation component of the Note Writing module. We selected high-performing models, including closed-source SoTA ones like Opus, Sonnet-3.5 and GPT-4-32k, alongside prominent open-source models with medical and general applications.

Datasets: We use two datasets for our evaluation. 1. Aci-bench (subTask B) (wai Yim et al., 2023): This is a public dataset designed for benchmarking automatic medical visit note generation. From this we take 39 different medical visits for our test set. 2. OncNoteGen Test: We choose a set of 21 oncology transcripts from OncNoteGen dataset such that it ensures coverage across criteria such as visit type (new vs. follow-up), length (long vs. short), and style (dictation-heavy vs. conversational). This is our proprietary dataset and is not available on the internet. On this particular dataset we also perform human expert based evaluation.

Human Scoring: To facilitate a rigorous assessment, human experts prepare rubrics which represent the gold-standard of the medical (oncology specific) entities (key phrases) which should be captured along with their respective sub-categories. These experts also create the gold-standard final notes designed to mirror the expectations of healthcare providers accurately.

Potential Leakage into Test Data: We recognise that it is possible that the Aci-bench data could have been present in the training sets of all the models that we compare against and also our base model - Mistral 7B. Even though we feel it is more likely to be present in the closed source models as compared to the smaller open-source models there is no way for us to know. In this framework we are guaranteed that the OncNoteGen Test Dataset is completely blind to the model by the virtue of it being entirely proprietary.

4.2.1 Information Extraction

Setup: We evaluated three tasks within the Oncology Entity Identification Task on the OncNoteGen Test dataset:

- **Missing Information**: We compared the generated note to the gold-standard note, assessing any missed phrases or key information, crucial for ensuring note coverage.
- **Redundant Information**: We identified redundant details in the generated note that were absent from the gold-standard, including "hallucinations" or unsubstantiated entities from transcripts, to maintain note conciseness and accuracy.
- **Misclassification**: We examined whether correctly identified entities were properly categorized, ensuring structured and well-organized notes.

Results and Analysis: Table-1 demonstrates that our domain-specific fine-tuning outperformed leading models like GPT-4-32k, particularly in reducing Missing Information, and was competitive in other tasks. Sonnet-3.5's improved performance highlights the value of leveraging recent datasets and better instructional comprehension, suggesting future opportunities. Our experts noted challenges like separating labs, biopsies, and imaging categories in the note, indicating areas for further tuning. Opus and Sonnet models experienced example leakage, reducing robustness, while models like Mistral, Llama, and BioMistral generated excessive redundant entities, impacting precision. Despite BioMistral's misleading high score in misclassification due to entity repetition, our model outshone the Mistral 7B Instruct base model, underscoring the benefits of specialty fine-tuning.

4.2.2 Note Writing Quality

Setup: We froze all SpecialtyScribe components, using our Information Extractor, and replaced the Note Writer with different LLMs, ensuring consistent input. Evaluations were conducted on both datasets described earlier.

Metrics: We used reference-based metrics like BLEU and ROUGE, which are common for summarization but have limitations in correlating with human judgment on creative tasks. Thus, human experts also assessed notes based on Clarity, Grammar, Professionalism, and Coherence.

Human Evaluation Methodology: Experts rated each note across the four parameters mentioned and used a 0–5 Likert Scale with scores normalized between 0 and 1. The final results were the sum of score across the 4 categories and reported for the OncNoteGen dataset.

Model Choice: Due to cost, we used GPT-4o instead of GPT-4-32k. Its claimed superiority makes it a strong benchmark. BioMistral was excluded for failing to follow output format instructions.

Results and Analysis: Table-2 indicates closer scores on Aci-bench compared to OncNoteGen. Our model surpassed both open and closed models, partly due to its understanding of the input style, showcasing the benefit of a custom-trained model. The higher performance gap on OncNoteGen highlights the limitations of generic models for specialized writing tasks. Notably, OncNoteGen's average scores were higher, attributed to prompts designed for a data distribution similar to that dataset.

4.2.3 Medical Note Generation

Setup To assess the overall impact of using SpecialtyScribe to generate medical notes, we compared the notes generated by various LLM's taking in the entire transcript with our framework as outlined in Section-3. We use the same metrics as defined in the previous task, except for human experts which now evaluate the note on multiple aspects.

Human Evaluation Methodology: The experts were asked to score the notes based on the following 4 verticals - *Writing Quality* (as explained in above task). *Clinical Accuracy* to determine how accurately the note reflects the original information from the medical encounter, including correct documentation of terms, findings, diagnoses, and treatment plans. *Completeness* to evaluate whether the note contains all necessary and relevant medical information without leaving any gaps in the patient's story or care and *Organization* to check the structure of the note, including accurate classification into medical sections. We follow a similar process as for Note Writer, where the experts are asked to give a score on the Likert scale between 0 to 5, which is then divided by 5 to get a number between 0 to 1 for each vertical. The final reported score is the sum of the scores for the 4 catergories averaged across the test set. We do this only for the OncNoteGen dataset.

Results and Analysis As indicated in Table-3, similar to values for the note quality evaluation task we see the model scores on Aci-bench dataset are not very different between the state of the art LLMs and our model. The scores on these metrics are also generally low as n-gram matching may simply require "heart murmur", but our prompts are structured to prompt the model to deliver full sentences like "Patient presents today for a consultation on heart murmurs". On OncNoteGen dataset, we can clearly see the superiority of our approach over the latest open source models. We perform on par with the latest models from Anthropic, falling slightly short of OpenAI's GPT-4o. Our human experts reported that our framework performed best in Writing Quality and Organization of the note. Even though Opus and GPT-4o models had the best coverage, they really struggled with note organization.

4.2.4 Ablation

To further substantiate the importance of every component in our framework, we conducted the

Model	Aci-bench (subTask B)		OncNoteGen		
	ROUGE_L	BLEU	ROUGE_L	BLEU	Human(4)
Opus	0.21	0.09	0.24	0.12	2.97
Sonnet-3.5	0.21	0.10	0.24	0.13	2.94
GPT-4o	0.18	0.07	0.21	0.10	**3.28**
mistralai/Mistral-7B-Instruct-v0.2	0.12	0.04	0.16	0.07	2.77
meta-llama/Meta-Llama-3-8B-Instruct	0.16	0.07	0.18	0.08	2.65
SpecialtyScribe (ours)	**0.24**	**0.12**	**0.31**	**0.21**	**3.17**
(w/o Context Retriever)	0.23	0.09	0.30	0.19	3.07
(w/o IE and Context Retriever)	0.24	0.11	0.29	0.18	2.51

Table 3: Results on Medical Note Generation Task (higher is better)

medical note generation experiment using two variations of the system. The first version removed the Context Retriever module, leaving the Note Writer model to rely solely on the Information Extractor model's output. In the second version, we eliminated both the Information Extraction and the Context Retriever modules, resulting in the Note Writer directly generating the end notes from the original input transcript. Table-3 clearly illustrates how each module of SpecialtyScribe framework is crucial for achieving optimal performance.

5 Conclusion

In this paper, we detail our efforts in creating a framework to generate medical specialty notes that can be adapted across multiple specialties. We train an Information Extraction (IE) model to extract medically relevant content from oncology-based doctor-patient conversations, develop a hallucination detection mechanism, and train a Note-Writer module to produce clinician-approved medical notes. Through rigorous evaluation, our findings reveal that our models and pipeline not only outperform the leading medical and general open-source models in this domain but also parallel the performance of the foremost proprietary models available. The results further demonstrate that decomposing the note generation task into smaller, manageable parts enhances both the accuracy and comprehensiveness of the medical notes produced. This approach ensures a more precise and reliable documentation system, which could significantly improve diagnostic and treatment practices in specialized medical care. Furthermore, our approach is cost-effective, achieving comparable performance to the most expensive models, such as Opus and GPT-4-32K, with a significantly smaller model.

Our work presents a framework that can serve as a foundation for further research to improve the automated medical note creation process, especially for complex medical specialties, potentially reducing clinician workloads.

6 Ethical Considerations

In compliance with HIPAA regulations, we have established Business Associate Agreements (BAAs) with OpenAI and Anthropic, the parent company of the Opus and Sonnet-3.5 models, to ensure the protection and confidentiality of sensitive data. This agreement guarantees that the data provided is neither leaked nor used for model training purposes. We thoroughly de-identified all personal health information (PHI) from our datasets before any processing or analysis. This was achieved by substituting PHI with non-identifiable entities using Named Entity Recognition (NER) techniques. Furthermore, the use of the SpecialtyScribe tool is strictly confined to internal operations for generating medical notes. To uphold ethical standards, we conduct regular audits of all input prompts to prevent any potential unethical usage.

7 Limitations

Future work should aim to construct and train a specialized embedding model to improve the detection and elimination of data hallucinations, thereby enhancing system accuracy and dependability. This paper primarily examines the framework in one specialty, yet there is ample opportunity to extend this research to include additional specialties, which would enhance the utility of the findings and the model's robustness across various fields. There is also potential for further advancements in both IE and summarizer models. Moreover, it's important to acknowledge that open-source datasets may not always mirror real-world complexities, underlining the need for publicly available datasets that can drive progress in this field.

8 Business Considerations

The scope of this work has been limited to protect the company's intellectual property (IP) and represents research-specific efforts. It does not directly reflect the exact models, architecture, or methods used in the company's production systems.

References

Lochan Basyal and Mihir Sanghvi. 2023. Text summarization using large language models: a comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models. *arXiv preprint arXiv:2310.10449*.

Sara Berg. 2023. 3 ways to begin to reduce clinical documentation by 75% by 2025. American Medical Association.

Anjanava Biswas and Wrick Talukdar. 2024. Intelligent clinical documentation: Harnessing generative ai for patient-centric clinical note generation. *arXiv preprint arXiv:2405.18346*.

Mikaël Chelli, Jules Descamps, Vincent Lavoué, Christophe Trojani, Michel Azar, Marcel Deckert, Jean-Luc Raynier, Gilles Clowez, Pascal Boileau, and Caroline Ruetsch-Chelli. 2024. Hallucination rates and reference accuracy of chatgpt and bard for systematic reviews: Comparative analysis. *Journal of Medical Internet Research*, 26:e53164.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation. *Preprint*, arXiv:2306.15595.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023b. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Sagar Goyal, Eti Rastogi, Sree Prasanna Rajagopal, Dong Yuan, Fen Zhao, Jai Chintagunta, Gautam Naik, and Jeff Ward. 2024. Healai: A healthcare llm for effective medical documentation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1167–1168.

Som Gupta and Sanjai Kumar Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *Preprint*, arXiv:1909.06146.

Tobias Kerner. 2024. Domain-specific pretraining of language models: A comparative study in the medical field. *arXiv preprint arXiv:2407.14076*.

Kundan Krishna, Sopan Khosla, Jeffrey P Bigham, and Zachary C Lipton. 2020. Generating soap notes from doctor-patient conversations using modular summarization techniques. *arXiv preprint arXiv:2005.01795*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. 2016. Clinical relation extraction with deep learning. *International Journal of Hybrid Information Technology*, 9(7):237–248.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason. *Preprint*, arXiv:2311.11045.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering. *Preprint*, arXiv:2203.14371.

Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. 2021. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):86.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C De Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the association for computational linguistics*, 2:143–154.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerova, et al. 2023. Clinical text summarization: adapting large language models can outperform human experts. *Research Square*.

Sassan Ghassemzadeh Vivek Podder, Valerie Lew. 2022. *SOAP Notes*. StatPearls Publishing, Treasure Island (FL).

Wen wai Yim, Yujuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. 2023. Aci-bench: a novel ambient clinical intelligence dataset for benchmarking automatic visit note generation. *Preprint*, arXiv:2306.02022.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. 2024. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, page ocae045.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. *Preprint*, arXiv:2309.12284.

Dong Yuan, Eti Rastogi, Gautam Naik, Jai Chintagunta, Sree Prasanna Rajagopal, Fen Zhao, Sagar Goyal, and Jeff Ward. 2024. A continued pretrained llm approach for automatic medical note generation. *arXiv preprint arXiv:2403.09057*.

Longxiang Zhang, Renato Negrinho, Arindam Ghosh, Vasudevan Jagannathan, Hamid Reza Hassanzadeh, Thomas Schaaf, and Matthew R Gormley. 2021. Leveraging pretrained models for automatic summarization of doctor-patient conversations. *arXiv preprint arXiv:2109.12174*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

A Detailed Implementation of Information Filtering algorithm

The Information Filter algorithm refines the output of the Information Extractor step (denoted as I) by returning a filtered subset that contains only information strongly aligned with the transcript. This process is crucial for mitigating hallucinations and ensuring the extracted information remains reliable.

We start with indexing the transcript by chunking it into variable-length sentence groups (e.g., 1, 2, 5 sentences) and computing their embeddings (E_T). Then, the extracted information (I) is decomposed into discrete items ($i_1, i_2, \ldots$) based on bullet points or new lines.

Step 1: Initial Matching Against Transcript For each decomposed item i, if it appears verbatim in the transcript, it is automatically included in the filtered set, denoted as $I_{included}$. However, if no exact match is found, the embeddings of the decomposed item are extracted, and a similarity score is computed against the transcript chunks. The most relevant transcript context is identified based on this score. To ensure reliability, any decomposed item with a similarity score below a predefined confidence threshold (θ) is filtered out. The threshold θ is domain-specific. In the medical field, it is kept low to ensure that any relevant information is not mistakenly discarded, even if it is phrased differently. This adjustment accounts for cases where the Information Extractor paraphrases content using medical terminology, such as converting "high blood pressure" to "hypertension."

Step 2: Secondary Filtering to Mitigate Hallucinations While a low threshold (θ) prevents the omission of important information, it may also allow irrelevant or hallucinated content to pass through. To further refine the selection, a second filtering step is applied. A similarity confidence score, denoted as α, is chosen empirically. Two embedding similarity scores are then computed. The PromptScore measures the similarity between the extracted information and the examples used in the prompt of the Information Extractor. The TranscriptScore measures the similarity between the extracted information and the input transcript. If the PromptScore exceeds α, while the TranscriptScore remains below α, the information is classified as a hallucination originating from the prompt and is removed. This step ensures that the extracted information is not overly influenced by the prompt ex-

amples and remains true to the original transcript. By systematically applying these steps, the Information Filter algorithm enhances the accuracy and reliability of extracted information, ensuring that medical notes are trustworthy, well-grounded in the original transcript, and free from hallucinations.

B Oncology Information Extraction Task Prompt

System
You are a highly trained and skilled AI medical doctor who specializes in writing a part of the Subjective section of a clinical SOAP (Subjective, Objective, Assessment, Plan) note. You only speak MARKDOWN.

User
<template>
{rules}
</template>

NOTE: If you are unsure or don't have enough information to provide a confident answer, do not create or imagine a response. Simply return "no information found". If a certain note template section lacks the necessary information within the transcript to be written, then leave that section blank.
<example>
Examples only for formatting reference.
For example: Let's say you want to write the sections CANCER PROCEDURES and CANCER SYMPTOMS from a given template. If no information is found related to CANCER PROCEDURES, the output should look like:

#CANCER PROCEDURES
##no information found
#CANCER SYMPTOMS
##<information here>
</example>

Using above template, example and guidelines, given the real transcript below, can you fill out the outline accurately and thoroughly? Return your answer as a string following the template. DO NOT return ANYTHING outside of the template.

Transcript:
{transcript}

C Additional Task Prompt

We utilized the GPT-4 model to generate question-answer pairs specific to certain sub-sections including 'Medications' and 'Plan-Orders', wherein the model initially encountered challenges. Beyond the generation tasks for general and respective sub-sections, we incorporated additional QA tasks that require short responses, with the aim to enhance the comprehension capabilities of the model

System
You are a medical assistant that can answer questions form a given context. In this task, you will be asked to answer a question from a given doctor patient transcript.
User
Transcript: {transcript}
Question: {question}
Return your response as a JSON in the following format:
{
"Answer": "....",
"Explanation": "...."
}