Agentic AI Architectures for SOAP Note Generation

Keno Hanken
Independent Researcher
Berlin, Germany
keno-hanken@labs.arg-os.eu

Abstract

Clinical documentation places significant time demands on medical professionals, consumes institutional resources, and is prone to errors that may compromise patient care. Recent advances in LLMs offer promising approaches for automating clinical note generation; however, the impact of different AI architectural designs remains underexplored, particularly for agentic AI systems. This study compares three architectures — single-LLM, multi-agentic, and swarm-agentic — for automated SOAP (Subjective, Objective, Assessment, Plan) note generation from doctor–patient dialogues. All approaches employ QLoRA-finetuned Ministral 3 models (3B and 8B parameters) trained on the MedSynth dataset (Mianroodi et al., 2025), comprising 10,030 dialogue–note pairs across 2,006 ICD-10 code classes. Performance is evaluated using ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore against a lexical-overlap baseline (dialogue vs. ground-truth SOAP, no inference). Results show that all finetuned models substantially outperform the baseline, while differences between architectural variants remain marginal. The single-LLM setup achieves the strongest performance across all metrics; 3B and 8B variants perform nearly identically on semantic similarity (BERTScore), while ROUGE differences are small but statistically significant. Qualitative inspection further reveals that residual differences across architectures are driven primarily by shared dataset priors rather than by architectural reasoning capacity. The results are based on synthetic data without human evaluation and reflect architectural behavior only.

1 Introduction

Clinical documentation serves multiple essential purposes including care continuity, research, and legal protection, but places significant time demands on medical professionals. The average consultant physician in clinical environments spends 12 hours a week adding to clinical documentation for patient records. This expenditure equals a productive monetary loss of 54,500 USD per year, per physician (Ignetica Ltd, 2022). Furthermore, 74% of medical professionals state that current efforts for clinical documentation impede patient care (American Medical Informatics Association, 2024). These findings suggest that a reduction of necessary efforts for patient records could address these and related challenges. However, a persisting challenge in documentation is the accurate and comprehensive documentation of relevant information. Medical professionals spend on average over an hour extra per day researching patient information (Ignetica Ltd, 2022). Adding to this, Bell et al. (2020) found in an empirical US study that 1 in 5 patient records contain at least one mistake, of which 40% are considered serious from the affected patients' perspective. Given these challenges, it is vital to make note-taking and information retrieval more time- and cost-effective.

Advances in LLM techniques offer a promising technical response to those challenges. There are countless strategies and process elements where AI can be used to enhance the process of note-taking, from transcribing speech, to reasoning about appropriate therapies for patients. Yet, the most impactful step is to ensure effective and accurate note composition from available data, such as transcripts. A widely adopted format for such notes is the SOAP schema (Subjective, Objective, Assessment, Plan), which structures clinical observations into four standardized dimensions and is detailed in Section 2.1. However, multiple strategies exist for how an AI infrastructure could solve the problem of composing clinical records, i.e. a single but powerful LLM could be employed to compose documentation, or a swarm of small, highly specialized AI agents could dissect the task of note-taking among them and collaboratively reason about the note composition. As for AI swarms, the landscape

Proceedings of the 25th Workshop on Biomedical Language Processing (BioNLP 2026), pages 742–752

of empirical studies in the health-care domain is still an emerging field of research. This study aims to serve as an orientation for AI-architectural solutions for the task of clinical note-taking, answering the question of *"Do multi- / swarm-agent architectures outperform a single finetuned LLM for SOAP note generation?"*. The contributions of this paper are:

- A systematic evaluation of single-LLM, multi-agent, and swarm-agentic architectures for clinical SOAP note generation under controlled conditions.
- Highlighting that increased architectural complexity does not yield meaningful performance gains, questioning the practical utility of agentic approaches for this task.
- A reproducible experimental setup for SOAP note generation based on the MedSynth dataset, including consistent training and evaluation procedures.

The remainder of this paper is structured as follows: Section 2 shows the academic background, Section 3 describes the experimental setup and model architectures, Section 4 presents the results, Section 5 discusses the findings, and Section 6 concludes.

2 Literature

2.1 Theoretical Background

Weed (1964) proposed guidelines for the composition and structure of medical records. Each one of the 5 rules addresses a distinct viewpoint of the available facts and assumptions during a medical consultation. The framework prescribes (1) review and organization of all available data sources for a holistic patient view; (2) interpretation of those data in light of the clinician's impression of the patient; (3) derivation of a treatment plan through inductive reasoning over the contextualized data; (4) continuous data acquisition aligned with the evolving treatment plan; and (5) explicit articulation of the relationships between data entities (Weed, 1964). Through this, Weed defined core principles of data-driven patient care. His data organization framework laid the groundwork for the now widely adopted SOAP note schema.

Podder et al. (2023) lay out the current widely used adoption of the SOAP schema. A SOAP note is composed of 4 dimensions, where:

- S stands for *Subjective*, capturing what the patient reports themselves (chief complaint, history of present illness, etc.).
- O stands for *Objective* and captures data obtained through observation or empirical measurement (i.e. ECG, RR, observed skin condition, etc.).
- A stands for *Assessment* and concerns the analysis of the S and O findings to deductively state a diagnosis and/or clarify existing differential diagnoses.
- P stands for *Planning* and lays out how a patient's condition is treated methodically and / or further monitored and assisted. (Podder et al., 2023)

Bommarito et al. (2025) define the criteria for an agentic system, by comparing successive levels of systems that process and act on data. A level 1 agentic system is defined as *"[...] any entity that pursues goals through perception and action, with at least minimal discretion over which action to take in response to what it perceives."* (Bommarito et al., 2025) meaning a system that possesses a *goal* to achieve, a *perception* of the environment, and the *actionable capabilities* to affect their environment. An example could be a thermostat which maintains a certain level of temperature through temperature measurements. It is important to notice that a level 1 agent's properties are not limited to the described properties, but also can be extended by one or more of the 3 additional properties, that a level 2 adds. An agentic system is considered to be level 2 if the system's logic follows specific *rules* that are deterministic and non-flexible. It extends level 1 agent capabilities by possessing a (continuous) *iterative* data / environment observation, an *adaptation* to challenges and new observations, and a *termination* under defined criteria. An example could be a web scraper that monitors news websites for specified keywords and notifies subscribers. A level 3 agentic system however leverages AI architectures like neural networks, LLMs, or VLMs with traditional computing resources, while possessing the same properties as a level 2 agent. A level 3 agent, for example, can be a trading bot that dynamically adjusts strategies based on government announcements and news. Bommarito et al. clarify that the definition of an agent is bound to, but not limited to, a system's capabilities to influence its environment

in different levels with different methods. Furthermore, the authors advocate for a clear definition of the term *agent*, as the term itself does not reflect the different tasks and properties an intelligent system possesses (Bommarito et al., 2025).

2.2 Related Work

Li et al. (2025) introduce an expert curated simulated dataset called *CliniKnote* of 1,200 doctor–patient conversations with clinical notes. For these notes, the SOAP schema was extended by a keyword section on top, creating the K-SOAP schema, to enable quick identification of relevant symptoms and diseases. Furthermore, the authors benchmarked the K-SOAP note generation across 22 LLMs via metrics such as ROUGE, BERTscore, Bleurt (Sellam et al., 2020), and QuestEval (Scialom et al., 2021). For keyword extraction and generation, a combination of clinical named entity recognition and relation extraction for symptoms and diseases from dialogues has been used. The authors defined 8 entity types combining 5 relational prefixes with symptom and disease categories. If a conversational contribution was identified to be relevant for one of the specified categories, the information was tagged. The benchmark was conducted across all tested LLMs in the 3 categories: baseline models, open-source LLMs, and commercial models. For open-source and baseline models, the authors used QLoRA to reduce resource demands, training adapters for finetuning in three modes: a single adapter for the full note, 4 adapters for grouped sections, or 15 separate adapters for each individual section (denoted with a *section-4* / *section-15* suffix). Commercial models were instead evaluated via zero-shot and one-shot in-context learning. The results showed that the GPT-4o model in one-shot mode (zero-shot was tested as well for closed-source models) outperformed all tested open source models and baseline models marginally in 7 out of 9 metrics, compared to the next best performing model qCammel-13b-section-15, a model optimized for academic medical knowledge. Among the baseline and open-source LLMs, the model qCammel-13b-section-15 outperformed other models moderately in 7 out of 9 metrics, compared to the second best performing model Llama3-8b-section-4 (Li et al., 2025). These results suggest that the type and quality of domain adaptation matter more than model size - as evidenced by OpenBioLLM's poor performance despite its biomedical specialization. However, GPT-4o's superior one-shot results indicate that scale remains a contributing factor.

Ramprasad et al. (2023) analyzed an approach for generating SOAP notes based on conversational data. The authors dissected the SOAP dimensions beyond the 4 well-known dimensions, down to the contained scopes of information each dimension holds. For example, the S-dimension holds information about the chief-complaint, past medical history, allergies, etc. In total, the study features 12 subdimensions. In their approach, the authors compared the performance of a general-purpose model (GPT 3.5) in zero- and few-shot settings, against a default finetuned BART-model, and a BART-model with modified architecture. The authors also compared 3 architectural modifications of the BART-model to support the identification of SOAP-specific information from the conversation input. **BART+EMB** adds SOAP-(sub-)section tokens to the BART-embedding. It was finetuned with annotated and tagged data, which provides explicit conditioning on the target SOAP section/subsection to guide summary content and style. **BART+ADAPT** integrated adapter modules - small bottleneck feed-forward networks - into each transformer layer, adopted from (Bapna and Firat, 2019). To enable SOAP note generation, full finetuning was applied. **BART+CA** adds a second cross-attention (CA) layer after the standard CA layer in each decoder block. While the default CA layer shares parameters across all sections, this additional layer uses separate parameters for attending to SOAP information. This enables the model to re-attend over the encoder output with a section-specific focus. These parameters are randomly initialized and learned during finetuning. Among the BART architectures, the BART+CA combination outperformed other modifications by a range of just 0.4–5.5 points in 3 of 4 applied metrics (ROUGE-L, UMLS(F-1), Isin). Compared to the baseline BART and GPT 3.5, the BART+CA modification outperformed other models in 2 of 4 metrics (UMLS(F-1), Direc(F-1)) by a range of 0.7 - 15.6 points (Ramprasad et al., 2023). These findings imply that a specialization on an architectural level for a task can benefit a model's performance.

3 Methodology

Figure 1 shows the methodological process steps.

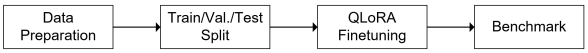

Figure 1: Methodology process steps

3.1 Dataset and Data Preparation

This study employed the MedSynth dataset, created by Mianroodi et al. (2025). The authors leveraged data from multiple sources, such as filed insurance claims from the IQVIA PharMetrics Plus dataset, and existing dialogue datasets, such as Aci-Bench (Yim et al., 2023), NoteChat (Wang et al., 2024), and PriMock57 (Papadopoulos Korfiatis et al., 2022). Despite the highlighted categorical skewness of the distribution of ICD-10 codes, the authors have selected a uniform sampling method, resulting in ca. 2,000 selected ICD-10 codes for English dialogue-note pair generation. The dialogues were synthetically generated with multiple AI agents, employing GPT-4o with Chain-of-Thought and In-Context Learning. The incident descriptions from the selected insurance claims were taken as an input for the GPT4o model, in accordance with the instruction for generating a dialogue. Through this, ca. 10,000 dialogue-note pairs were created, where each ICD-10 code is generally represented 5 times.

The dataset version retrieved in Feb. 2026 comprises 10,240 dialogue-note pairs in total, featuring 2,037 ICD-10 code classes. After cleaning and preparation, the final dataset comprised of 10,030 records in 2,006 ICD-10 code classes. For the later training of multi- and swarm-agentic AI systems, the SOAP notes have been split via regex in their 4 respective dimensions. The evaluation dataset was constructed by randomly (all random selections employ a fixed seed) selecting one sample per ICD-10 code, resulting in 2,006 test instances (20%). This ensures uniform code-level coverage in the final evaluation. The remaining samples are randomly shuffled split into training and validation sets. Specifically, 802 samples (8%) are used for validation to support model selection, while the remaining 7,222 samples (72%) are used for training. This splitting strategy balances the need for representative evaluation across clinical categories with maintaining sufficient training data for reliable finetuning.

The records from the test set, serving as ground-truth SOAP notes, were scored directly against their source dialogues to establish a lexical-overlap baseline. All notes were run directly against their source dialogues without model inference, quantifying the inherent textual distance between raw dialogue and structured SOAP output. This baseline is intentionally conservative: it does not represent a competitive note-generation system, but rather provides a lower-bound reference that quantifies how much information must be reorganized, paraphrased, and inferred to transform raw dialogue into a structured SOAP note. Direct numerical comparison with dedicated baselines from prior work such as Ramprasad et al. (2023) or Li et al. (2025) is informally enabled through the shared use of ROUGE and BERTScore, although differences in datasets, target schemas, and reference quality preclude strict equivalence.

3.2 Benchmark Metrics

The following metrics were applied:

- **ROUGE**: This metric was proposed by Lin (2004) and stands for *Recall-Oriented Understudy for Gisting Evaluation*. This metric evaluates the overlap between generated and reference texts using n-gram matching. ROUGE-1 and ROUGE-2 measure unigram and bigram overlap respectively, while ROUGE-L captures sentence-level structural similarity without requiring consecutive n-gram matches. The metric reports precision, recall, and F1, all of which were reported.

- **BERTScore**: BERTScore was proposed by Zhang et al. (2020). BERTScore computes cosine similarity between contextual embeddings of tokens in the generated and reference texts, enabling it to capture semantic equivalence even when different words are used. This study leverages the RoBERTa-large model (Liu et al., 2019) to compute the score. Furthermore, the metric reports precision, recall, and F1, of which all were reported.

The combination of ROUGE and BERTScore provides complementary evaluation perspectives: ROUGE captures lexical overlap through surface-level n-gram matching, while BERTScore assesses semantic similarity through contextual embeddings, recognizing paraphrases that ROUGE would miss. Both metrics are widely adopted in clinical note generation research. This enables a direct comparison with prior work such as Li et al. (2025); Mianroodi et al. (2025) and others.

Recent shared tasks on clinical text generation, such as RRG24 in Xu et al. (2024), have employed broader metric suites including F1-RadGraph and F1-CheXbert for radiology report generation. These metrics are specifically tailored to radiology and rely on radiology-domain labelers, whereas the *Discharge Me!* subtask instead used metrics such as MEDCON and AlignScore for discharge summary generation. Neither suite transfers directly to general SOAP note evaluation across the full ICD-10 spectrum, where no analogous domain-specific labelers are established. ROUGE and BERTScore remain the metrics most consistently reported in the SOAP-generation literature and enable direct comparison with prior work (Li et al., 2025; Ramprasad et al., 2023).

3.3 LLM Selection and Finetuning

3.3.1 LLM selection

All architectures in this study employed the Ministral 3 model series (Liu et al., 2026). Specifically, the 3B and 8B reasoning model versions *mistralai/Ministral-3-3B-Reasoning-2512* and *mistralai/Ministral-3-8B-Reasoning-2512* were used for finetuning and benchmarking in this study. The model family was published by Mistral AI in Jan. 2026. Its strong performance on the MMLU-Redux benchmark (Gema et al., 2025), a benchmark for multitask language understanding where LLMs are confronted with questions from all kinds of domains, made the model an ideal choice. Additional consideration was given to its native support for structured output generation and level 3 agentic AI capabilities. The model employs a decoder-only transformer architecture and supports context lengths of up to 256k tokens; sufficient for processing medical dialogues with generative instructions.

3.3.2 Quantization

To still leverage the model's capabilities while improving memory efficiency, the quantization technique *Quantized Low Rank Adaptation* (QLoRA) was used. Introduced by Dettmers et al. (2023), QLoRA combines *Normal Float 4* (NF4) quantization with *Low Rank Adaptation* (LoRA) (Hu et al., 2022), enabling parameter-efficient finetuning by keeping base weights frozen and training small low-rank adapter matrices instead. It further reduces memory usage through double quantization.

Quantization represents model weights in lower-precision formats, trading off compression and accuracy. While Ministral 3 natively uses *bfloat16*, alternatives such as INT8, INT4, FP8, and NF4 offer varying memory savings. Given the focus on maintaining quality under strong compression, NF4 was selected.

QLoRA exploits the near-normal distribution of pretrained weights by aligning quantization levels with weight density, minimizing information loss at 4-bit precision. Double quantization further compresses the per-block scaling factors, reducing overhead to $\sim$0.127 bits per parameter. During training, only LoRA adapters are updated while 4-bit weights are temporarily dequantized to bfloat16; paged optimizers stabilize memory usage on single-GPU setups.

This study applied LoRA hyperparameters as listed in Table 1:

Parameter	Value	Description
Rank (r)	64	Dimension of the low-rank matrices
Alpha (α)	32	Scaling factor for LoRA updates
Dropout	0.05	Dropout probability
Target modules	q/k/v/o/gate/ up/down_proj	Transformer layers adapted
Bias	None	No bias params trained
Task type	CAUSAL_LM	Language modeling objective

Table 1: LoRA hyperparameter configuration

Preliminary runs with r=16 and r=32 showed marginally lower validation performance; r=64 was selected as the best tradeoff.

3.3.3 Finetuning

The finetuning of the respective base models with 3B and 8B parameters was conducted on the training split with 7,222 observations, and tested against the validation split of 802 observations for each epoch. Table 2 shows the applied hyperparameters for training for all architectures.

The finetuning results suggest that the models reach their global optimum within 3 epochs. Figure 2 displays the performance of the training and validation for the single-LLM architecture, which

Parameter	Value
Learning rate	2×10^{-4}
Epochs	3
Per-device batch size	2
Gradient accumulation steps	4
Effective batch size	8
Optimizer	AdamW
Learning rate scheduler	Cosine
Warmup ratio	0.05
Weight decay	0.01
Max sequence length	4096
Gradient checkpointing	Enabled
Precision	bf16

Table 2: Finetuning hyperparameter configuration

can be considered representative for other models, employed in different architectures, as other finetuning runs show similar results. Training was conducted on 3× NVIDIA A100-40GB GPUs.

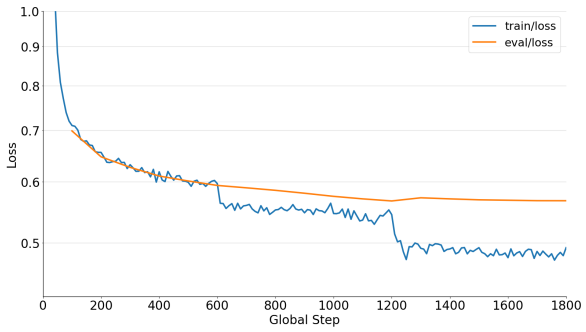

Figure 2: Single-LLM setup 8B model training and validation performance

3.3.4 Preservation of Base Model Capabilities

A potential concern when finetuning models for downstream agentic use is the loss of general instruction-following capability that pre-trained models acquire during instruction tuning. The QLoRA setup adopted here was specifically chosen to mitigate this risk. Three properties of the configuration are relevant: (1) All base model weights remain frozen in 4-bit NF4 quantization throughout training; gradient updates are applied only to the rank-64 LoRA adapter matrices, which account for ca. 2.9% (3B) and ca. 2.3% (8B) of total parameters — a range consistent with standard QLoRA configurations applied to comparable base-model scales (Dettmers et al., 2023; Li et al., 2025) (2) Each adapter parameterizes its weight update as a low-rank decomposition ($\Delta W = BA$, with rank r = 64), restricting the modification of every targeted projection to a 64-dimensional subspace; this structural constraint bounds how far the adapted layer can deviate from the frozen pre-trained projection. (3) During inference, agents continue to receive role-specific system prompts (defining role, expected output structure, and clinical scope) — finetuning thus complements, rather than replaces, prompt-based agent specialization. Empirical support for this design choice can be found in prior clinical NLP work that combines QLoRA adapters with structured prompting for SOAP-related tasks (Li et al., 2025). An alternative design — for example, orchestrating off-the-shelf instruction-tuned LLMs through prompts alone — represents a valid and complementary configuration, and is positioned as a future-work direction in Section 6.

3.4 Agentic AI Architectures

Before the architectures are described, it is worth noting that for both the multi-agentic and swarm-agentic AI architectures, trained LoRA adapters were applied via hot-swap to a single base model during inference, significantly reducing VRAM demands as only one base model needs to be held in memory. For all architectures, all LLMs were either equipped with 3B or 8B parameters.

Note that this study employs role-specialized adapters within a fixed pipeline rather than prompt-orchestrated agents over a single instruction-tuned base model. Coordination between agents is structural — information flow is hard-wired through the architecture (Figures 4, 5) — not negotiated at inference. This design isolates architectural decomposition as the experimental variable while holding base model and training recipe constant; alternative prompt-driven setups represent a complementary design space (see Section 6).

3.4.1 Single-LLM Architecture

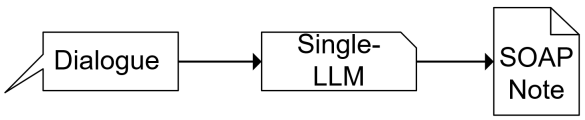

Figure 3: Single-LLM architecture visualized

The single-LLM setup featured a finetuned model, trained to compose an entire SOAP note. During training, the entire SOAP note for the individual case was visible. This enabled the model to capture the holistic format and structure of the SOAP-schema. For inference, the trained adapter

was applied. The purpose of the single-LLM setup is to serve as a controlled comparison against agentic AI architectures with multiple AI agents. Figure 3 visualizes the single-LLM architecture.

3.4.2 Multi-Agentic AI Architecture

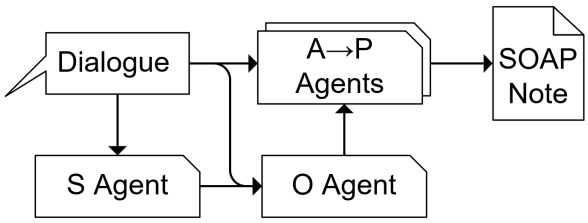

Figure 4: Multi-agentic AI architecture visualized

The multi-agentic AI setup featured 4 agents in sequential order, each one finetuned in their respective domain. Depending on their relative position in the order of the SOAP schema, the models receive the dialogue, as well as the output of the previous AI agents. For the final output, all dimension-outputs were concatenated in the appropriate SOAP order. Figure 4 visualizes the multi-agentic AI architecture.

For training, the SOAP notes have been separated in their respective domains, to control the access of information for each SOAP stage. Every domain-respective agent received a specialized system prompt (e.g., defining role, expected output format, and relevant clinical concepts) to ensure dimension-specific information sensitivity for phrases and concepts described in the dialogue.

3.4.3 Swarm-Agentic AI Architecture

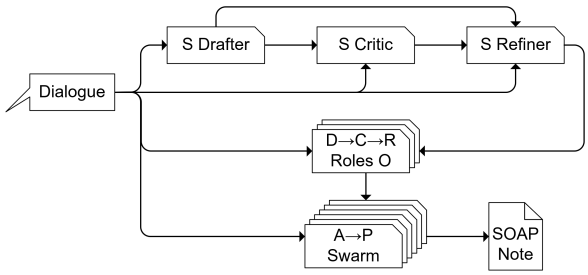

Figure 5: Swarm-agentic AI architecture visualized

The swarm-agentic AI setup featured 12 agents in sequential order, with 3 agents per SOAP dimension acting as drafter, critic, and refiner. For every dialogue input, the *drafter* composes a SOAP note for the respective dimension. The *critic* then judges the drafter's output against the original dialogue, based on structure and criteria learned through finetuning. The *refiner* refines the draft based on the dialogue, drafter output, and critic output, and outputs the final SOAP-dimension note. All dimension outputs were concatenated in the appropriate SOAP order. Figure 5 visualizes the swarm-agentic AI architecture.

During training, each of the 12 agents received a specialized system prompt (e.g., defining role, expected output format, and relevant clinical concepts) to ensure dimension specialization through role understanding and sensitivity to relevant phrases and concepts. The drafter was trained on dialogue-dimension pairs with the ground-truth SOAP dimension as gold text, analogous to the multi-agent training. The critic relied on synthetic data: a cross-paired imperfect draft is created by swapping the SOAP dimension with one from a different consultation sharing the same ICD-10 code, and a programmatic function generates a synthetic critique comparing that draft against the ground-truth SOAP note and current dialogue. The critic thus learns to generate a critique from dialogue and imperfect draft. The refiner is then trained on the same cross-pairs, receiving dialogue, imperfect draft, and synthetic critique as input, with the gold-standard SOAP dimension as target. For inference, the 12 trained adapters were applied via hot-swap to the same base model.

4 Results

4.1 Quantitative Results

For benchmarking, the test set with 2,006 observations, representing one ICD-10 code class each, was employed. The computed quality metrics allowed a per-case review for every setup and every agent. Table 3 features the combined performance metrics over all architecture respective agents, where the top performing architectural metric is formatted in **bold**, while the 2nd best performing architecture metrics are underlined.

The single-LLM architecture achieves the strongest performance, ranking first across all metrics, with the 8B variant marginally outperforming the 3B variant. Both share nearly identical BERTScores ($\Delta = 0.002$). The multi-agent and swarm-agent setups follow closely, with Multi 8B consistently outperforming Swarm 8B across all metrics, while Single 3B holds the second-ranked position in all four metrics. The performance differences across all finetuned architectures are marginal, with the largest gap being 0.042 in ROUGE-L between Single 8B (0.628) and Multi

	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Baseline	0.461	0.227	0.326	0.825
Single 3B	0.771$_{\pm.002}$	0.543$_{\pm.003}$	0.615$_{\pm.003}$	0.928$_{\pm.001}$
Single 8B	**0.778**$_{\pm.002}$	**0.554**$_{\pm.003}$	**0.628**$_{\pm.003}$	**0.930**$_{\pm.001}$
Multi 3B	0.753$_{\pm.003}$	0.519$_{\pm.003}$	0.586$_{\pm.004}$	0.921$_{\pm.001}$
Multi 8B	0.760$_{\pm.002}$	0.530$_{\pm.003}$	0.598$_{\pm.003}$	0.922$_{\pm.001}$
Swarm 3B	0.752$_{\pm.003}$	0.522$_{\pm.004}$	0.591$_{\pm.004}$	0.920$_{\pm.001}$
Swarm 8B	0.754$_{\pm.003}$	0.526$_{\pm.003}$	0.594$_{\pm.004}$	0.921$_{\pm.001}$

Table 3: Test set F1 results. Baseline denotes lexical-overlap on ground-truth SOAP notes. ±values denote 95% bootstrap CI half-widths (10,000 resamples).

3B (0.586).

To evaluate statistical significance, paired t-tests and Wilcoxon signed-rank tests were conducted on per-case scores across all 2,006 test instances, with Cohen's d as effect size. The single-LLM architecture significantly outperforms both the multi-agent and swarm-agent setups across all metrics (all $p < 0.001$), with medium BERTScore effect sizes (d = 0.67–0.80 for 8B; d = 0.65–0.70 for 3B). The multi-agent vs. swarm-agent gap is negligible (BERTScore Δ = 0.0007, d = 0.050; Wilcoxon p = 0.161). For model scale, 8B variants show small but significant ROUGE improvements over 3B ($d \approx 0.15$–0.28), while BERTScore differences exhibit negligible effect sizes for both multi- and swarm-agent settings ($d \leq 0.060$; p = 0.149 and p = 0.007 respectively), suggesting that architectural choice dominates over parameter count in practical terms.

4.2 Qualitative Analysis

A stratified inspection of 100 generated notes, drawn uniformly across ICD-10 categories, revealed three recurring patterns across all finetuned configurations.

Demographic hallucinations in the Subjective dimension. All architectures frequently produced patient demographic attributes (age, gender, ethnicity) absent from the source dialogue. These attributes appear consistently in MedSynth's reference notes without corresponding dialogue evidence, and models reproduced them as a learned prior rather than through factual extraction — a dataset-level bias that architectural decomposition did not mitigate.

Empty Objective dimensions. Approximately 15% of reference notes contained no Objective content, reflecting MedSynth's simulated telemedicine scenarios. While all architectures correctly reproduced empty sections in these cases, multi- and swarm-agent variants occasionally hallucinated brief filler text ("examination not performed", "vitals not recorded"), marginally reducing reference alignment.

Error propagation in sequential pipelines. In the swarm setup, errors introduced at the drafter stage — such as fabricated medication names — tended to persist through critic and refiner stages. The critic, trained on synthetic cross-paired critiques, produced plausible critique text but failed to catch factual issues requiring external clinical knowledge. This aligns with the negligible BERTScore gap between multi- and swarm-agent variants (Section 4.1) and explains why additional refinement stages did not yield gains.

Together, these observations suggest that performance differences across architectures are driven less by architectural reasoning capacity than by shared dataset priors and drafter error propagation — limitations unlikely to be resolved by further architectural complexity.

5 Discussion

The results show that overall the finetuned models substantially outperform the lexical-overlap baseline, which measures the inherent textual overlap between raw dialogues and ground-truth SOAP notes without any model inference. This confirms the value of task-specific finetuning.

Figure 6 shows the F1 BERTScore delta between architectures; among finetuned models, the absolute maximum is 0.008 and the minimum is 0 — indicating that 3B models perform competitively with larger and more complex configurations alike. A plausible explanation is that SOAP note generation, as a structured reorganization and paraphrase task, does not expose the complexity that multi-agent decomposition is designed to exploit: a single finetuned model captures the full dialogue-to-

Δ	Base	Single 3B	Single 8B	Multi 3B	Multi 8B	Swarm 3B	Swarm 8B
Base		-0.103	-0.105	-0.096	-0.097	-0.096	-0.096
Single 3B	0.103		-0.002	0.007	0.006	0.008	0.007
Single 8B	0.105	0.002		0.009	0.008	0.009	0.009
Multi 3B	0.096	-0.007	-0.009		-0.001	0.001	0.000
Multi 8B	0.097	-0.006	-0.008	0.001		0.002	0.002
Swarm 3B	0.096	-0.008	-0.009	-0.001	-0.002		0.000
Swarm 8B	0.096	-0.007	-0.009	0.000	-0.002	0.000	

Figure 6: Delta for F1 BERTScore benchmark results comparison (row - column)

note mapping end-to-end, without the coordination overhead and error propagation inherent to pipeline architectures. We therefore interpret the flat architectural performance not as a failure mode but as an empirical answer — agentic decomposition is not warranted for this class of tasks. This counterpoints the broader trend of defaulting to multi-agent orchestration in clinical NLP and motivates the targeted question outlined in Section 6: identifying the specific task properties — multi-specialty, longitudinal, multi-modal — under which decomposition can be expected to pay off.

6 Conclusion

This study compared three agentic AI architectures — single-LLM, multi-agentic, and swarm-agentic — for automated SOAP note generation from doctor–patient dialogues, using finetuned Ministral 3 models (3B and 8B) on the MedSynth dataset. All finetuned architectures substantially outperformed the lexical-overlap baseline, underscoring the value of task-specific finetuning in clinical documentation. Performance differences among the three approaches were minimal, however, with the single-LLM setup leading in all 4 metrics. These findings suggest that increasing architectural complexity through multi-agent or swarm decomposition does not provide meaningful gains over a single finetuned model for this task. Notably, the 3B and 8B variants performed nearly identically, indicating that smaller models can handle structured clinical NLP tasks without compromising quality.

These findings carry practical implications: organizations seeking to automate clinical documentation can achieve competitive results with a single, compact finetuned model, reducing both infrastructure complexity and computational cost compared to multi-agent deployments.

Several limitations should be acknowledged. The evaluation relied exclusively on automated metrics (ROUGE, BERTScore), which may not fully capture clinical accuracy or factual correctness — dimensions that require expert human evaluation. Furthermore, the study was conducted on a single synthetic dataset, and generalizability to real-world clinical dialogues with noise from speech recognition remains to be validated.

As detailed in Section 4.2, the inference outputs reveal dataset-induced behavior — demographic hallucinations in the Subjective dimension and reproduction of empty Objective sections from MedSynth's telemedicine scenarios — that limits its practical utility independently of architectural choice, since key content decisions appear driven by pattern reproduction, not factual grounding.

Future work should (1) extend evaluation to authentic clinical transcripts with acoustic and disfluency noise; (2) incorporate human expert assessment of factual correctness, completeness, and clinical utility; (3) test additional base-model families and instruction-only orchestrations to disentangle architectural from base-model effects; (4) explore broader, factuality-oriented metrics (e.g., RRG24-style entity-level scores adapted to SOAP content (Xu et al., 2024)); and (5) investigate whether more complex clinical tasks — multi-specialty consultations, longitudinal patient records, or multi-modal inputs combining dialogue with imaging or lab data — provide the conditions under which agentic decomposition begins to pay off.

7 Acknowledgements

This paper is dedicated to my loving partner Anna-Isabel Steffens for her unwavering belief in me.

My sincere gratitude to my project partner, Julia Jellinek, for her outstanding support in data engineering.

References

American Medical Informatics Association. 2024. TrendBurden-results-april-2024.

Ankur Bapna and Orhan Firat. 2019. Simple, Scalable Adaptation for Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Sigall K. Bell, Tom Delbanco, Joann G. Elmore, Patricia S. Fitzgerald, Alan Fossa, Kendall Harcourt, Suzanne G. Leveille, Thomas H. Payne, Rebecca A. Stametz, Jan Walker, and Catherine M. DesRoches. 2020. Frequency and types of patient-reported errors in electronic health record ambulatory care notes. *JAMA Network Open*, 3(6):e205867.

Michael James Bommarito, Jillian Bommarito, and Daniel Martin Katz. 2025. What is an agent? a conceptual primer and history of agents and agentic AI.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLORA: Efficient Finetuning of Quantized LLMs. *Advances in neural information processing systems*, 36:10088–10115.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, and 1 others. 2025. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ignetica Ltd. 2022. Assessing the burden of clinical documentation.

Yizhan Li, Sifan Wu, Christopher Smith, Thomas Lo, and Bang Liu. 2025. Improving clinical note generation from complex doctor-patient conversation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 209–221. Springer.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, and 1 others. 2026. Ministral 3. *arXiv preprint arXiv:2601.08584*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ahmad Rezaie Mianroodi, Amirali Rezaie, Niko Grisel Todorov, Cyril Rakovski, and Frank Rudzicz. 2025. Medsynth: Realistic, synthetic medical dialogue-note pairs. *Preprint*, arXiv:2508.01401.

Alex Papadopoulos Korfiatis, Francesco Moramarco, Radmila Sarac, and Aleksandar Savkov. 2022. PriMock57: A Dataset Of Primary Care Mock Consultations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 588–598, Dublin, Ireland. Association for Computational Linguistics.

Vivek Podder, Valerie Lew, and Sassan Ghassemzadeh. 2023. SOAP notes. In *StatPearls*. StatPearls Publishing.

Sanjana Ramprasad, Elisa Ferracane, and Sai P Selvaraj. 2023. Generating more faithful and consistent SOAP notes using attribute-specific parameters. *Proceedings of the 8th Machine Learning for Healthcare Conference*, 8(219):631–649.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. Questeval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6594–6604.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7881–7892.

Junda Wang, Zonghai Yao, Zhichao Yang, Huixue Zhou, Rumeng Li, Xun Wang, Yucheng Xu, and Hong Yu. 2024. NoteChat: A Dataset of Synthetic Patient-Physician Conversations Conditioned on Clinical Notes. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15183–15201, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Lawrence L. Weed. 1964. Medical records, patient care, and medical education. *Irish Journal of Medical Science*, 39(6):271–282.

Justin Xu, Zhihong Chen, Andrew Johnston, Louis Blankemeier, Maya Varma, Jason Hom, William J Collins, Ankit Modi, Robert Lloyd, Benjamin Hopkins, and 1 others. 2024. Overview of the first shared task on clinical text generation: Rrg24 and "discharge me!". In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 85–98.

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. *Scientific Data*, 10(1):586.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.