

POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering



GETALP-MISTRAL7B: A CLINICAL LARGE LANGUAGE MODEL FOR AUTOMATED DISCHARGE DOCUMENTATION FROM ELECTRONIC HEALTH RECORDS

Supervisors

Prof. Didier SCHWAB

Prof. Lorraine GOEURIOT

Prof. Gabriella OLMO

Candidate

Michele PANTALEO

October 2025

Abstract

This work presents GETALP-Mistral7B, a clinical large language model designed to automatically generate discharge documentation. Leveraging patients’ Electronic Health Records (EHRs), the model generates two central sections of a discharge summary: the hospital course and the discharge instructions.

EHRs are usually stored in structured formats that differ across hospitals [1, 2, 3, 4, 5, 6, 7]. To ensure interoperability across heterogeneous systems, EHRs were transformed into two task-specific textual documents: the diary for generating the hospital course, the patient summary for producing the discharge instructions.

GETALP-Mistral7B is fine-tuned from Asclepius-Mistral7B [8] using encounters from the Beth Israel Deaconess Medical Center (MIMIC-IV [9]). Quantized low rank adaptation (QLoRA) [10] is used to fine-tune the model separately for each section, yielding two specialized lightweight adapters while keeping the base model weights frozen.

GETALP-Mistral7B is benchmarked against models from the first shared task on clinical text generation: *Discharge-Me!* [11, 12]. Evaluation is conducted using the challenge’s framework, which consists of a held-out set of 250 examples and eight NLP metrics assessing lexical similarity, semantic adequacy, and factual correctness. GETALP-Mistral7B achieves an overall score of 0.393, establishing it as the state-of-the-art for generating discharge documentation.

Table of Contents

Acronyms	VI
Glossary	IX
1 Introduction	1
1.1 Context and Motivation	2
1.1.1 Regulatory requirements	2
1.1.2 Timeliness and impact on patient outcomes	3
1.1.3 Length, usability, and outpatient provider needs	3
1.1.4 Physician workload and administrative burden	4
1.1.5 Divergent Views on Discharge Summaries	4
1.2 Challenges	5
1.2.1 Task Complexity	5
1.2.2 Timeliness and Workflow Integration	6
1.2.3 Data Quality and Ground Truth Limitations	6
1.2.4 Privacy and Security Concerns	7
1.2.5 Evaluation	7
1.2.6 Acceptance and Trust	8
1.3 Objectives	8
1.3.1 Textual Representation of EHRs	9
1.3.2 Lightweight, Modular, and Instruction-Tuned LLM	10
1.3.3 Abstractive Text Summarization and Controlled Generation	10
1.4 Contributions	12
1.4.1 Electronic Health Record Processing	12
1.4.2 GETALP-Mistral7B	13
1.5 Reminder	14
2 State of the Art	16
2.1 Overview of LLMs in Medicine	16
2.1.1 Biomedical vs Clinical Text	16

2.1.2	Natural Language: Understanding vs Generation in Clinical Models	17
2.2	Online LLMs	18
2.3	Open Source Models	19
2.3.1	Asclepius	19
2.4	Discharge-Me! BioNLP ACL’24 Shared Task on Streamlining Discharge Documentation	20
2.4.1	Discharge-Me! Results	21
2.4.2	Limitation of the <i>Discharge-Me!</i> Solutions	23
2.4.3	Deploying to Overcome Limitations	23
3	Methods	24
3.1	Source Documents	25
3.1.1	Diary	25
3.1.2	Patient Summary	28
3.2	Source Document Generator	29
3.2.1	LogBuilder	30
3.2.2	DiaryMaker	31
3.3	Prompt Assembler	32
3.4	MIMIC-IV	32
3.4.1	<i>ED</i> Module	33
3.4.2	<i>HOSP</i> Module	33
3.4.3	<i>NOTE</i> Module	35
3.4.4	Other Modules: <i>ICU</i> , <i>CXR</i> , <i>ECG</i>	40
3.4.5	MIMIC-IV Diary	40
3.4.6	MIMIC-IV Patient Summary	42
3.4.7	Data Split: Train, Test, DB, Recollection	43
3.4.8	MIMIC-IV Source Document Length	46
3.5	Training GETALP Models	47
3.5.1	Supervised Fine-Tuning	47
3.5.2	QLoRA Fine-Tuning	47
3.6	Evaluation Framework	48
3.6.1	Metrics	48
3.6.2	Per-Task Evaluation	53
3.6.3	Overall Scoring System	53
3.7	Experiments	53
3.7.1	Evaluation of GETALP Models	54
3.7.2	Ablation Study on the Diary Content	54
3.7.3	Using AI-Generated Hospital Courses for Discharge Instruction Generation	56
3.7.4	Qualitative Analysis of Model Outputs	57

3.7.5	Recollection Test	57
4	Results	58
4.1	Evaluation of GETALP Models	58
4.1.1	GETALP vs Asclepius	58
4.1.2	GETALP-Mistral7B vs GETALP-Llama3-8B	59
4.1.3	GETALP vs <i>Discharge-Me!</i>	60
4.2	Ablation Study on the Diary	61
4.3	Using AI-Generated Hospital Courses for Discharge Instruction Gen- eration	62
4.4	Qualitative Analysis of Model Outputs	63
4.4.1	Hallucinations	64
4.4.2	Degeneration	64
4.4.3	Redundancy	68
4.5	Recollection Test	68
5	Conclusions	70
5.1	Feasibility	70
5.2	Source Document Design and Components	71
5.2.1	Design	71
5.2.2	Components	72
5.3	Investigating Degeneration	73
5.4	Evaluation Framework: A Review	73
5.5	Privacy and Recollection	74
6	Towards an End-to-End Discharge Summary Automation	76
6.1	History of Present Illness, Social Background, Family History	77
6.2	Patient Medical History	78
6.3	Non-Textual Clinical Documents	78
A	Examples	80
A.1	Prompts	81
A.2	Source Documents	82
A.3	Reference and Inference	85
A.3.1	Hospital Course	85
A.3.2	Discharge Instructions	86
B	MIMIC-IV: Statistics	87
B.1	Number of events per admission	87
B.2	Text Length	88

C	Technical Details	97
C.1	Model Architectures	97
C.2	Training Details	98
C.2.1	Parameters	98
C.2.2	Memory Usage and Learning Curves	99
C.3	Inference Details	100
C.4	Hardware	100
C.5	Software	101
C.6	Environment	101
C.6.1	Training Phase	101
C.6.2	Inference Phase	102
C.6.3	Total Emissions	102
D	Metrics	103
D.1	Candidate, Reference, n-grams, counting, F1-score	103
D.2	BLEU	104
D.3	ROUGE	104
D.3.1	ROUGE-N	104
D.3.2	ROUGE-L	105
D.4	METEOR	105
D.5	BERTScore	106
D.6	AlignScore	106
D.7	MEDCON	107
	Bibliography	108

Acronyms

AI artificial intelligence

API application programming interface

ASR automatic speech recognition

AVG per-task average score

BIDMC Beth Israel Deaconess Medical Center

BHC brief hospital course

BLEU bilingual evaluation understudy

CP care provider

CXR chest x-ray

CT computed tomography

DES dynamic expert selection

DI discharge instructions

DS discharge summary

ECG electrocardiogram

ED emergency department

EHR electronic health record

EOS end of sequence

FH family history

GETALP Groupe d'Étude en Traduction Automatique/Traitement Automatisé
des Langues et de la Parole

HC hospital course

HEENT head, eyes, ears, nose, and throat

HIPAA Health Insurance Portability and Accountability Act

HF Hugging Face

HPI history of present illness

ICD international classification of diseases

ICU intensive care unit

ICP inpatient care provider

LAD left anterior descending (artery)

LCS longest common subsequence

LLM large language model

LoRA low rank adaptation

METEOR metric for evaluation of translation with explicit ordering

MLP multi layer perceptron

MRI medical resonance image

NER name entity recognition

NLG natural language generation

NLI natural language inference

NLP natural language processing

NLU natural language understanding

OCP outpatient care provider

PCP primary care physician

PMH patient medical history

POD post-operative day

PS patient summary

QLoRA quantized low rank adaptation

ROUGE recall-oriented understudy for gisting evalaution

SB social background

SFT supervised fine tuning

SOAP subjective objective assessment plan

SD source document

UMLS unified medical language system

Glossary

Asclepius

A publicly shareable clinical language model developed to overcome the privacy constraints of real clinical notes. Trained on a large corpus of synthetic discharge-style notes derived from anonymized case reports, Asclepius demonstrates strong performance across diverse clinical tasks and supports task-specific adaptation. In this study, it is referred to as *base model* or *(clinical) foundation model* because it serves as the starting point for fine-tuning GETALP models. Specifically, Asclepius-Llama3-8B was fine-tuned to produce GETALP-Llama3-8B, and Asclepius-Mistral7B was fine-tuned to produce GETALP-Mistral7B.

Beth Israel Deaconess Medical Center

A major teaching hospital of Harvard Medical School in Boston, Massachusetts. It is the primary source of the clinical data found in the MIMIC-IV database.

care provider

A care provider (CP) is a physician, nurse practitioner, physician assistant, or other licensed healthcare professional responsible for managing and delivering patient care. In this study, we categorize CPs into two groups based on the patient’s care setting: ICPs, who provide care during hospital stays, and OCPs, who provide care when patients are not hospitalized (outpatient).

diary

A document deployed in this study, used as input to generate the hospital course for a given patient. It presents clinical events in chronological order, grouped by date. An example is provided in Appendix A.2.

discharge instructions

A section of the discharge summary intended to communicate to the patient (and their caregivers) critical information regarding post-hospital care. It

includes instructions on medication use, follow-up appointments, lifestyle changes, symptom monitoring, and when to seek emergency care. The goal of discharge instructions is to ensure that patients understand their care plan and adhere to post-discharge recommendations in order to reduce readmissions and promote recovery. It’s one of the two target section of this study.

Synonyms: post-care instructions

discharge summary

A narrative clinical document summarizing the key events, diagnoses, and care plan from an inpatient stay, intended to support seamless transitions to outpatient or post-acute care. The discharge summary plays a critical role in facilitating continuity of care and ensuring that important information is communicated between healthcare providers. It is divided into many sections. Two of them are the focus of this study: hospital course and discharge instructions

Discharge-Me!

The *Discharge-Me!* challenge is a shared task designed to reduce the time and effort clinicians spend on writing discharge summaries. Participants are asked to develop clinical AI-assistants to generate the brief hospital course and discharge instructions sections of BIDMC’s discharge summaries. This work is inspired by this challenge.

electronic health record

Digital version of a patient’s medical chart, containing comprehensive health information accessible to authorized healthcare providers. It’s a real-time, patient-centered repository that facilitates seamless information sharing and decision-making. EHRs are designed to be used by multiple care providers and healthcare organizations

GETALP-Mistral7B

The state-of-the-art model developed in this study, which excels in comparison to models from the *Discharge-Me!* challenge. It is fine-tuned to generate hospital course and discharge instructions from specialized input documents (diary and patient summary, respectively). The model is based on the Asclepius-Mistral7B, and was adapted using QLoRA-based instruction fine-tuning. The architecture includes two lightweight adapters, each dedicated to one of the two generation tasks. The name originates from the name for the research group which created it, GETALP, and the 7 billion parameter version of the Mistral.

GETALP-Llama3-8B

A model developed in this study, which performs better than GETALP-Mistral7B in predicting hospital course and does not suffer from text degeneration. Despite obtaining lower overall automatic scores compared to GETALP-Mistral7B, it still outperforms all other models from the *Discharge-Me!* challenge. Like its sibling, it is fine-tuned to generate hospital course and discharge instructions from specialized input documents (diary and patient summary, respectively). The model is based on the Asclepius-Llama3-8B foundation model, and was adapted using QLoRA-based instruction fine-tuning. The architecture includes two lightweight adapters, each dedicated to one of the two generation tasks. The name originates from the name for the research group which created it, GETALP, and the 8 billion parameter version of Llama3.

GitHub

A hosting service for version control and collaborative software development. The code of with this work is stored at github.com/pantaleo18/discharge-me. Access to the repository may be temporarily restricted, as the process of removing information protected by the PhysioNet DUA from the development Jupyter notebooks may still be ongoing at the time of reading.

hospital course

A part of the discharge summary that summarizes key events from admission to discharge, detailing all the events that characterized a patient’s inpatient stay. This includes significant diagnoses, treatments, complications, and any changes in the patient’s condition. HC is one of the two target section of this study.

Health Insurance Portability and Accountability Act

A United States law enacted in 1996 that provides data privacy and security provisions for safeguarding medical information. In this work, HIPAA compliance is critical for ensuring patient data remains secure and confidential when using automated systems.

Hugging Face

A collaborative platform for sharing AI models and datasets. Models and databases related to this work are stored on HF, but are not publicly available due to the PhysioNet DUA protecting MIMIC-IV. Website: huggingface.co

inpatient care provider

Inpatient Care Providers (ICPs) are CPs who oversee patient care during hospitalizations. ICPs are typically hospitalists, specialists, or residents. They are primarily responsible for preparing DSs that summarize the care delivered during the hospital stay.

MIMIC-IV

The Medical Information Mart for Intensive Care, fourth (IV) edition, is a large, publicly available database containing deidentified health-related data. The data originates from patients admitted to the intensive care units of the Beth Israel Deaconess Medical Center.

MIMIC-IV-ED

The ED module contains data from the Emergency Department. It captures patient triage, diagnoses, medications, vital signs, and emergency department stays. Each stay is identified by a unique `stay_id`. Tables (with *italic* indicating those used in this study): *edstays*, *medrecon*, triage, diagnosis, pyxis, vitalsigns.

MIMIC-IV-HOSP

The HOSP module contains data collected from the hospital-wide electronic health record (EHR). Tables (with *italic* indicating those used in this study): *admissions*, *patients*, *transfers*, *labevents*, *microbiologyevents*, *prescriptions*, *pharmacy*, *procedures_icd*, *poe*, *drgcodes*, *diagnoses_icd*, *services*, *hcupsevents*, *omr*, *emar*, *provider*, *poe_detail*.

MIMIC-IV-NOTE

The NOTE module stores unstructured clinical notes, mainly discharge summaries and radiology reports. Discharge summaries contain detailed accounts of hospitalizations, while radiology reports provide formal interpretations of imaging tests. Both are linked to hospital admissions via `hadm_id`. Tables (with *italic* indicating those used in this study): *discharge*, *radiology*.

outpatient care provider

Outpatient Care Providers (OCPs) are CPs who provide care when patients are not hospitalized (outpatient). OCPs are the main recipients of DSes prepared by ICPs to coordinate follow-up care. This group includes primary care physicians (PCPs) and other specialists managing patients in non-hospitalized settings.

primary care physician

Primary Care Physicians (PCPs) are a subset of OCPs, typically general practitioners or internists, who provide ongoing care for patients when they are not hospitalized. PCPs manage preventive care, chronic illnesses, and act as the main coordinators of patient care. They are often the principal readers of DSeS and play a key role in guiding post-discharge care.

patient summary

A document defined in this study, used as input to generate the discharge instructions. It includes information on allergies, medical history, hospital course, discharge medications, and conditions. An example is provided in Appendix A.2.

PhysioNet

A public repository of clinical and physiological datasets, widely used in medical research. Access to MIMIC-IV and other sensitive resources is regulated by a Data Use Agreement (DUA), which requires individual registration and completion of an ethics training. The DUA explicitly forbids publishing models, weights, or solutions trained on MIMIC-IV outside PhysioNet, and prohibits transferring MIMIC-IV data to external AI services or online platforms.

source document

A document used as input to GETALP models to generate a specific target section of a discharge summary. The type of source document depends on the target section: when generating hospital courses, the source document is a diary; when generating discharge instructions, the source document is a patient summary.

Chapter 1

Introduction

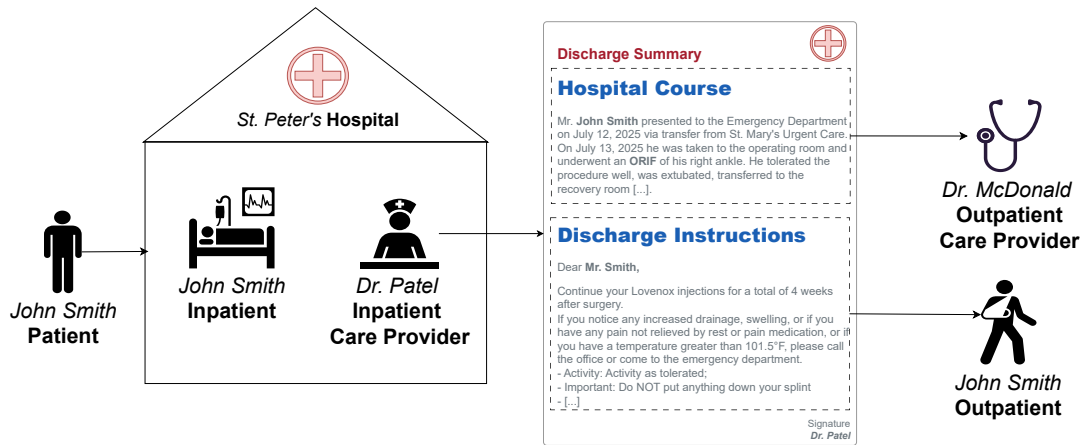


Figure 1.1: A discharge summary is the clinical document in object of this study. It serves as a comprehensive record of a patient’s hospitalization and is composed of multiple sections that each play a specific role in ensuring safe and effective care transitions.

In this figure, only two of its sections are shown: the **Hospital Course** and the **Discharge Instructions**, as these are the focus of the large language model developed in this work. Although being part of the same document, the two sections are intended for different audiences: an hospital course is primarily written for other clinicians, summarizing the diagnostic and therapeutic decisions taken during the hospital stay, while discharge instructions are directed toward a patients and their families, to support post-discharge self-care.

All names and identifiers shown in the figure are fictional. Appendix A shows an example of discharge summary.

1.1 Context and Motivation

The **discharge summary (DS)** is one of the most critical documents produced during the course of a patient’s hospitalization. It represents the primary communication tool between inpatient care provider (ICPs), responsible for the patient’s care during the hospital stay, and the outpatient care provider (OCPs), who will continue the patient’s care after discharge [13]. When well-prepared, the DS ensures continuity of care, enhances patient safety, and facilitates efficient transitions between settings.

In essence, a DS provides a comprehensive yet concise narrative of the hospitalization, including the reason for admission, the patient’s **hospital course (HC)**, final diagnoses, procedures performed, medications prescribed at discharge, follow-up plans, and specific **discharge instructions (DI)** for the patient and family [14]. Among its various sections, hospital course and discharge instructions present specific challenges for automatic generation: HC requires summarizing complex clinical events in a coherent narrative for professional readers, while DI must convey clear and actionable information for patients and families. **This study focuses on using AI methods to generate two key sections of discharge summaries: HC and DI.**

A detailed examination of the regulatory standards governing the style of discharge summaries is next provided, followed by an analysis of their timeliness and impact on patient outcomes. Subsequently, considerations related to length, usability, and the needs of outpatient providers are addressed, and the section concludes by examining the differing perspectives between inpatient and outpatient providers.

1.1.1 Regulatory requirements

The **Joint Commission (formerly JCAHO)** establishes specific regulatory standards regarding the content and completion of discharge summaries [15]. These standards mandate that a DS include at least six key sections:

1. reason for hospitalization,
2. significant findings and diagnoses,
3. **procedures and treatment provided** (i.e. the hospital course),
4. the patient’s condition at discharge,
5. **patient and family instructions** (i.e. the discharge instructions),
6. and the attending physician’s signature.

In addition to content requirements, JCAHO mandates that the discharge summary must be completed and placed in the patient’s medical record within **30 days of discharge or transfer**, and preferably as soon as possible after the event, so that outpatient providers can act on the information [15]. Among these sections, several have been consistently identified by clinicians as the **most important** for ensuring safe care transitions: **hospital course**, discharge diagnoses, medication reconciliation, and **discharge instructions** [16]. These sections are prioritized because they summarize the key clinical reasoning behind the hospitalization and provide actionable next steps for outpatient care. Despite these standards, compliance remains uneven due to systemic challenges.

1.1.2 Timeliness and impact on patient outcomes

Although critically important, discharge summaries are often delayed. Completing a DS at the time of discharge can take between **30 minutes and two hours**, depending on the complexity of the case [14]. Given competing clinical demands, many physicians postpone or delegate the writing of DS, which often results in the documents being finalized only several days or even weeks after discharge. Such delays have been consistently associated with adverse patient outcomes.

Li et al. demonstrated that the absence of a DS at discharge was associated with a **79% increase in the likelihood of readmission** within 7 days [17]. Moreover, the same work reported that if a DS was not completed within 7 days after discharge, the risk of readmission within 7–28 days was comparable to that observed when no summary was written at all. In other words, **failure to complete a DS within 7 days after discharge is essentially equivalent to not writing one at all**, in terms of readmission risk.

Similarly, Hoyer et al. reported that each additional 3-day delay in DS completion was associated with a **1% increase in the probability of readmission** [18]. Taken together, these findings highlight the critical importance of timely DS delivery for ensuring continuity of care and preventing unnecessary hospital readmissions.

1.1.3 Length, usability, and outpatient provider needs

Length and readability of discharge summaries heavily impact their utility. Primary care physicians (PCPs) often report having insufficient time to read lengthy documents; most review a DS for only **1–5 minutes** before a follow-up visit [16]. Therefore, they rely heavily on well-structured summaries and expect the most critical elements—hospital course, discharge diagnoses, medication reconciliation, and follow-up instructions—to be easy to locate and concise [16]. When these sections are incomplete, ambiguous, or buried within extraneous details, information can be missed, potentially resulting in lapses in care continuity.

1.1.4 Physician workload and administrative burden

The preparation of DSes further increases the already substantial burden on clinicians. Studies have shown that physicians spend roughly **twice as much time on indirect patient care** tasks as on direct bedside care [19, 20]. Indirect tasks include documentation and activities related to electronic health records (EHRs).

This imbalance contributes to job dissatisfaction and burnout, which are further exacerbated by frequent interruptions and the pressure to manage high patient volumes [21, 22]. In a mature EHR environment, the time-motion study by Momenipur and Pennathur revealed that clinicians often struggle to balance the need for thorough documentation with providing real-time patient care [22]. The requirement to complete DSes promptly can therefore feel burdensome, particularly when clinical workloads are high.

1.1.5 Divergent Views on Discharge Summaries

Divergent perspectives between the authors and the readers of the DS—namely, the ICPs and the OCPs—further complicate its preparation. A common perception among ICPs is that their **outpatient colleagues do not fully value the document** they produce. Conversely, OCPs frequently **report frustration with the quality and timeliness** of the discharge summaries they receive.

A large-scale survey conducted in the United States by Sorita et al. quantified these contrasting viewpoints [13]. The study compared the opinions of hospitalists, who are the predominant type of ICP and the primary authors of the DS, with those of PCPs, who are often the primary recipients of DSes. The data revealed a significant misalignment in perceptions:

- 44% percent of hospitalists believed they were “too busy to prepare a high-quality discharge summary,” a view shared by only 23% of the PCPs.
- 60% of hospitalists were of the opinion that “PCPs have insufficient time to read an entire discharge summary,” whereas only 38% of PCPs agreed with this statement.

These misaligned expectations between the authors and the audience of the DS are a critical barrier and likely contribute to the persistence of suboptimal documentation practices, even though both groups fundamentally agree on the importance of the document for a safe patient transition.

While these challenges are deeply rooted and multifaceted, advances in natural language processing (NLP) and large language models (LLMs) offer a promising path toward scalable solutions—provided key technical and clinical barriers can be addressed. The upcoming section introduces the challenges encountered while automating clinical documentation.

1.2 Challenges

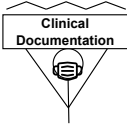
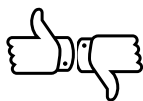
Intrinsic	 Burdensome	 Complex	 Timeliness
Ethical	 Social Acceptance	 Privacy	 Data Sharing
Technical	 Model Evaluation	 Data Leakage	 Data Quality

Figure 1.2: Challenges. This picture summarizes the challenges one may encounter while automatizing clinical documentation. While some challenge is intrinsic in clinical documentation (e.g burdensome, complexity, timeiliness), other challenges are either social (acceptance, privacy) or technical concerns (data quality, evaluation).

Automating the generation of discharge summaries could significantly reduce clinician workload and ensure that summaries are available in a timely manner. However, developing a robust automated system is a non-trivial endeavor. Several challenges must be addressed before such solutions can be safely adopted in clinical practice.

1.2.1 Task Complexity

One of the main challenges lies in the complexity of the content that must be captured:

- **Hospital Course:** this section provides a **chronological summary of the patient’s hospital stay**, condensing a long and complex sequence of clinical events into a coherent narrative. It involves selecting and integrating information from heterogeneous EHRs sources—such as progress notes, laboratory

results, radiology reports, and procedure logs—while filtering out incidental or non-contributory findings. The main challenge lies in exercising *clinical reasoning* to distinguish which events are truly relevant for understanding the patient’s trajectory, a task complicated by the frequent presence of abnormal values or minor findings whose significance is often uncertain [23, 24].

- **Discharge Instructions:** this section translates complex clinical information into clear, **patient-friendly instructions**. It must communicate critical guidance on medication adherence, activity restrictions, symptom monitoring, and follow-up care in layman’s terms, without sacrificing medical accuracy. Poorly written or ambiguous discharge instructions can lead to misunderstanding, non-adherence, and readmissions [25].

It may be worth to point out that **some sections of the discharge summary cannot be AI-generated**. For example, diagnoses and medication plan are doctor’s direct decisions. An intelligent system can’t guess what the physician has determined for the patient.

1.2.2 Timeliness and Workflow Integration

Another challenge is ensuring that automated solutions support, rather than disrupt, clinical workflows. DSes must be available at the time of discharge, but clinicians typically prioritize acute patient care over documentation, particularly in high-volume hospital settings. If an automated system requires additional steps or interrupts existing processes, its adoption is likely to be poor [22, 21].

Effective solutions must therefore integrate seamlessly into the clinical environment, ideally reducing the effort required to generate a DS while preserving its accuracy and completeness. Specifically, an AI-assistant for discharge documentation should leverage **raw clinical data** directly, thereby relieving clinicians from concerns about note style or formatting, and minimizing the time wasted navigating poorly designed EHR interfaces.

1.2.3 Data Quality and Ground Truth Limitations

Automated DS generation is heavily dependent on the quality of the input data and training sets. Unfortunately, real-world EHRs are often **noisy, repetitive, and inconsistent** across providers and institutions [26, 27]. Clinical notes can be incomplete, copied forward with minimal updates, or formatted in ways that obscure key information. Extracting accurate and meaningful content from these sources is a significant challenge.

In particular, DSes also suffer from quality issues. Studies have shown that these documents often contain errors, omissions, or stylistic inconsistencies that,

when used to train AI clinical assistants, can propagate into automated outputs if models simply learn to replicate them [23, 24]. For instance, diagnoses in DSEs were found to be inaccurate in up to 55% of summaries in one cohort study [23], and medication lists frequently omit critical details such as indications and reasons for medication changes [25]. These limitations complicate the development of reliable models.

1.2.4 Privacy and Security Concerns

Criminals actively target medical records because they contain demographic data, social security numbers, insurance details, and clinical histories that can be misused for **identity theft, fraud, unauthorized medical care**, prescription abuse, tax scams, and **blackmail**—thus offering far greater value than financial records, which are easier to block once compromised [28, 29]. Recent large-scale breaches—such as Kaiser (13.4 million affected), Cencora, and Harvard Pilgrim—underscore the systemic risks facing healthcare systems [30]. The Vastaamo psychotherapy scandal in Finland (33000 patients) revealed the devastating human toll when therapy notes were ex-filtrated and used for **extortion**. The attacker demanded 370K euros from the provider, then €200–500 per patient, contributing to **at least one suicide** and prompting new data-protection legislation [31, 32, 33].

Most state-of-the-art LLMs (e.g., ChatGPT, Gemini) are hosted on remote cloud servers and can be accessed only **online** through web-based applications or via APIs. Thus, employing these models entails **transmitting EHRs outside hospital** firewalls, thereby exposing patient data to the **risks and consequences** outlined above, including major privacy concerns and a concrete risk of non-compliance with HIPAA [34], GDPR [35], and the newborn EU AI-ACT [36, 20].

Consequently, automated DS generation systems must be designed for **fully offline, secure deployment within hospital infrastructures**, ensuring that no identifiable patient information leaves the premises. This requirement has important implications for model selection, optimization, and hardware constraints.

1.2.5 Evaluation

Automatic evaluation of AI-generated clinical reports is inherently **difficult**. Clinical text is semantically dense, highly specialized, and often allows for multiple equally valid formulations. Metrics based solely on **lexical overlap**, such as BLEU [37], ROUGE [38], or METEOR [39], fail to capture factual correctness or clinical appropriateness. More advanced metrics, such as BERTScore [40], AlignScore [41], and MEDCON [42], can better capture **semantic similarity** and **concept alignment**, but they cannot fully replace **human judgment** in assessing clinical correctness and relevance.

For these reasons, evaluation by clinical experts remains the **gold standard**, as only domain specialists can reliably assess factual accuracy, completeness, and whether the generated text truly reflects the patient’s hospital stay.

1.2.6 Acceptance and Trust

In addition to technical and workflow issues, a crucial barrier to adopting AI for clinical documentation is societal and professional acceptance.

Patients and the general public generally recognize the potential benefits of AI in healthcare but also express strong reservations. Gundlack et al. conducted qualitative interviews in Germany and found that while participants appreciated AI’s efficiency and potential to reduce clinician burden, they **feared loss of the “human touch”** in patient care and worried about errors and data privacy breaches [43]. Similarly, Young et al. reported that although many patients acknowledged improvements in diagnostic accuracy and system efficiency, a substantial proportion voiced concerns about **safety, transparency, and accountability** when AI was used in clinical decision-making [44].

Healthcare professionals demonstrate similar **ambivalence**. Hamedani et al. surveyed physicians and nurses in Iran and reported that 65% of respondents **supported AI** adoption and 84% believed it could bring positive change, yet 78% admitted they did **not fully trust AI outcomes**, and 39% expressed fears of **professional deskilling** [45]. Negash et al., through in-depth interviews with German physicians, found that clinicians were receptive to AI as a support tool but insisted on maintaining final authority over clinical decisions and documentation [46]. Likewise, Rony et al. showed that healthcare workers in a large cross-sectional study expressed interest in AI but highlighted a **lack of clear regulatory** frameworks and uncertainty about **liability** in case of AI errors [47].

Altogether, these findings suggest that both clinicians and patients emphasize that **AI should serve as an aid**, not a replacement, and that safeguards such as **explainability** must be prioritized while developing AI models for medicine.

1.3 Objectives

The objective of this study is **to develop an AI assistant to automate clinical documentation**. More specifically, attention is focused on two critical and time-consuming sections of a discharge summary: the Hospital Course and the Discharge Instructions.

To this end, the following goals are pursued:

1. Development of a **full-text representation of EHRs** that remains consistent and interoperable across hospitals, health data management systems, and

diverse data formats.

2. Development of a **lightweight, modular, instruction-tuned LLM** for discharge summary section-specific generation.

Also, framing the generative tasks in terms of **Abstractive Text Summarization** and **Controlled Generation** provides a clear methodological perspective and situates this work within the broader landscape of **NLP** research.

1.3.1 Textual Representation of EHRs

One of the key obstacles while dealing with the clinical domain is the heterogeneous nature of electronic health records. Public hospitals, private clinics, academic research centers, and nationalized healthcare systems all differ in how patient data is managed and structured. For example:

- In the **United States**, many hospitals rely on commercial **EHR vendors** (e.g., Epic, Oracle Health) that store clinical data in highly structured, often proprietary, relational databases [1, 2, 3];
- In **Europe**, there is an attempt to create a common European Health Data Space [48]. In countries such as **Italy** and **France**, healthcare systems are largely nationalized, with patient data managed through *Fascicolo Sanitario Elettronico* [4, 5, 6] and *Mon Espace Santé* [7], respectively. Despite these initiatives, heterogeneity persists due to differences in regional policies, procurement decisions, and levels of digitalization.

In practice, hospitals adopting different EHR systems operate on distinct data models. For example, Stanford Medicine maintains two Epic Clarity data models that are “similar but not identical”, each comprising more than 10,000 tables and supporting a wide range of data types, from encounters and laboratory results to various clinical notes [49, 50]. In the United Kingdom, Cambridge University Hospitals run Epic as their core electronic patient record system, while Milton Keynes University Hospital uses Oracle Cerner Millennium, originally deployed through the National Programme for IT [51, 52]. In the United States, the Department of Veterans Affairs is rolling out Oracle Cerner across its facilities [53], whereas other hospitals such as Medical City Dallas and Princeton Community Hospital rely on MEDITECH, with the latter reporting performance improvements after adopting MEDITECH Expanse [54, 55, 56]. **These systems differ** not only in vendor but also in the **size, structure, and organization** of their underlying schemas, as well as the balance between structured fields and textual data captured in the record.

The inherent diversity of raw clinical data makes the design of a *one-size-fits-all* EHR pre-processing system impractical. Therefore, a core objective of this study is to **restructure EHRs into task-specific textual documents**, since LLMs are generally designed to process textual data. This modular approach is effective in eliminating the tight coupling between the model and the database’s original format, which provides a significant advantage: it **makes the model’s input agnostic with respect to the EHRs’ original format**. This provides key benefits in terms of **portability**, as the same assistant can work across institutions regardless of their internal EHR formats; **maintainability**, since only the document generation step must be adapted when data formats change; and **extensibility**, which allows for future support for additional data types (e.g., images) to be added without retraining the model by transforming those data into textual document (e.g., through *captioning*).

1.3.2 Lightweight, Modular, and Instruction-Tuned LLM

The pipeline shall implement a **lightweight, modular, and instruction-tuned LLM** capable of generating documentation for specific sections of a discharge summary. A model with the aforementioned **features** allows for the meeting of the practical demands of real-world clinical environments:

- **Ensuring Timeliness:** The system’s compact size and efficient nature must guarantee low latency, allowing documentation to be generated promptly and integrated seamlessly into clinical workflows.
- **Adhering to Privacy Regulations:** The model must be small enough to deploy on-premise within a hospital’s firewall, ensuring that sensitive patient data remains compliant with regulations.
- **Respecting Resource Constraints:** By minimizing the model’s size and computational requirements, the system should reduce the need for significant hardware upgrades, making it more financially viable and accessible for hospitals.

1.3.3 Abstractive Text Summarization and Controlled Generation

To address the clinical documentation challenges, the work is framed within the NLP field. The generation of a hospital course is approached as a task of **abstractive summarization**, whereas the creation of discharge instructions as a task of **controlled generation**. This methodological choice provides a clear framework for the solutions, enabling the leveraging of established techniques and contextualizing the contribution within a broader research landscape.

Abstractive Summarization for Hospital Courses

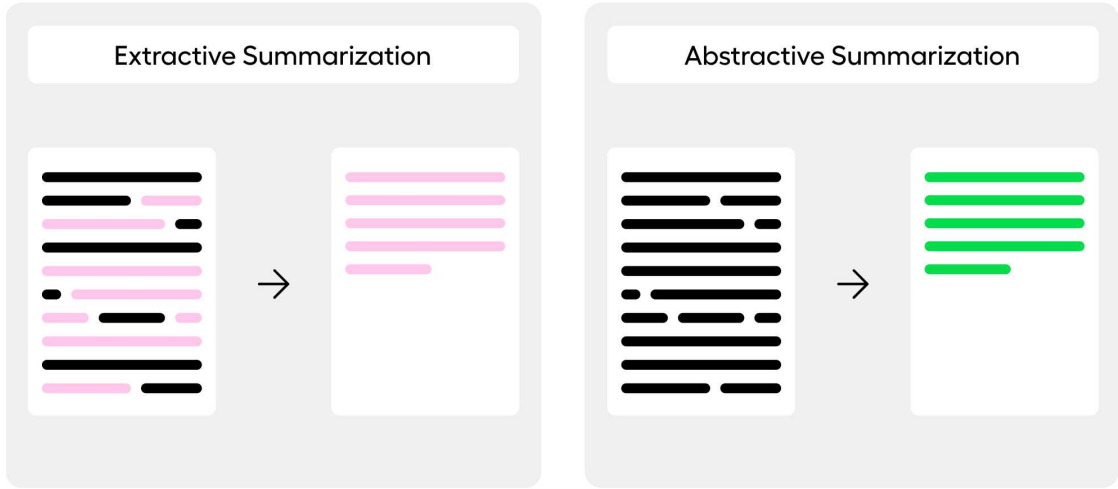


Figure 1.3: Extractive vs **Abstractive** summarization [57]. An extractive summary can use only those sentences that already exist in the original text. An abstractive summary, instead, generates new text by **paraphrasing** and **condensing** the original input. Abstractive summarization is used by the system to create a narrative description of a patient’s hospital stay (i.e., the hospital course) from a time-ordered list of clinical events.

Generating a hospital course requires abstractive summarization. This process goes beyond simply extracting sentences from the source material; it involves synthesizing information from various clinical events and structured medical records to create a new, coherent narrative. The goal is to produce a detailed yet concise summary that accurately reflects the patient’s hospitalization, written in the style of a clinical professional. This task requires the model to perform clinical reasoning to distinguish relevant information from noise and to structure it chronologically.

Controlled Generation for Discharge Instructions

Creating discharge instructions involve controlled generation. This task focuses on generating new text that adheres to specific constraints. The primary controls are the output’s clarity, style, and content. The generated instructions must be medically accurate while ensuring that they are clear, actionable, and easily accessible for a non-specialist audience, such as patients and caregivers. This requires the model to simplify complex medical terminology and structure the information in a patient-friendly format, ensuring all critical guidance on medications, follow-up, and self-care is communicated effectively.

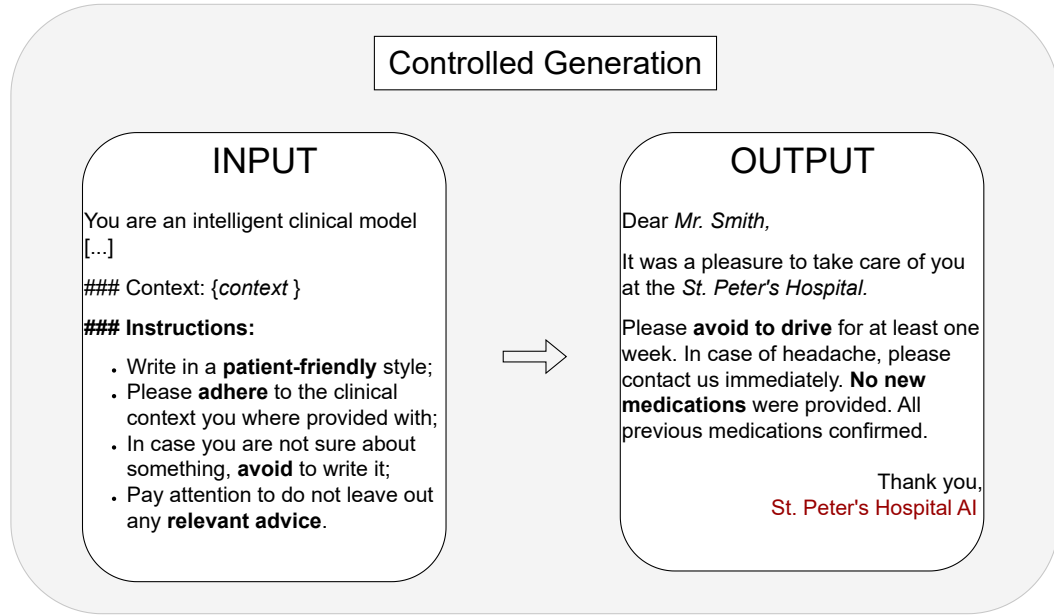


Figure 1.4: Controlled Generation. The model receives clear instructions that guide both its content and style. These instructions are also designed to prevent hallucinations, fictional outputs, and harmful content.

1.4 Contributions

This work proposes a **pipeline for automating the generation of discharge summary** sections, focusing on the hospital course and discharge instructions (Figure 1.5). The contributions, detailed below, fulfill the objectives set in Section 1.3.

1.4.1 Electronic Health Record Processing

As discussed in 1.3.1, hospitals differ widely in how patient data is stored and managed, making it impractical to design a clinical AI-assistant that interfaces directly with heterogeneous EHR databases. A modular approach solves this by introducing a standardized representation of the clinical context: once the relevant data has been reorganized into this format, the model can operate without regard to the original database schema or storage method. This representation is broadly referred to as the **Source Document**, a task-specific textual document in which clinical data—originally stored across multiple tables, formats, and modalities—is processed and merged into a single text file. Two distinct types of source documents are defined, each tailored to its own task:

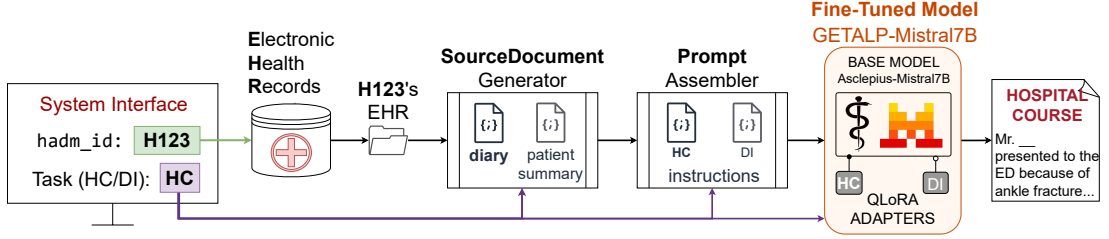


Figure 1.5: The System. Given a hospital admission identifier (`hadm_id`) and the target section to be generated—either hospital course or discharge instructions—the system first retrieves the patient’s EHRs from the hospital database. These EHRs are then processed and structured into a textual document, referred to as the source document, that provides all the facts necessary to accomplish the requested task. In this example, EHRs are converted into a diary—a chronological log-like file tailored for HC production. The source document is combined with task-specific instructions and passed to the AI assistant, **GETALP-Mistral7B**, which is trained on top of **Asclepius-Mistral7B**—a clinical LLM derived from the general-purpose **Mistral7B**. To support section-specific generation, two lightweight QLoRA adapter modules were trained: one for generating hospital courses and another for discharge instructions. The system outputs the requested section of the DS; in this case, the hospital course. Examples of such documents can be found in Appendix A.

- **Diary:** a chronological log of clinical events, grouped by date. It is used as clinical context for the **generation of hospital courses**;
- **Patient Summary:** a document grouping relevant information into specific sections (e.g., allergies, medical history, hospital course, discharge medications, and conditions). It is used as clinical context for the **generation of discharge instructions**.

While the building of a patient summary only requires to stitch together some stay’s info, including the hospital course itself, a longer data processing is required to craft the diary. Examples of both documents are provided in Appendix A.2.

It may be worth to further stress that this approach allows the same model to work across hospitals without modification: only the Source Document Generator needs to be adapted to convert local health records into diary/patient summary formats—**no model retraining or any further system changes are required**.

1.4.2 GETALP-Mistral7B

GETALP-Mistral7B is a clinical LLM specialized to write hospital courses and discharge instructions. **GETALP-Mistral7B** is created by **fine-tuning** the first

publicly sharaeble clinical LLM **Asclepius-Mistral7B-v0.3** [8] over **MIMIC-IV**, which is a publicly available, de-identified EHRs database containing a vast amount of data from patients who were admitted to the emergency department or intensive care units at Beth Israel Deaconess Medical Center (BIDMC) [9].

Given the limited computational resources typically available in hospital settings [58, 59], Quantized Low-Rank Adaptation (**QLoRA**) is employed for quantization aware fine-tuning. This approach generates two specialized **lightweight adapters**, one for HCs and another for DI. These compact adapters can be injected into the base model as needed, **maintaining a small memory footprint while avoiding catastrophic forgetting** of its original medical knowledge. The resulting GETALP-Mistral7B model can be loaded in 4-bit format during inference, allowing its operation on modest hardware and making it suitable for deployment in hospital environments.

The model is named after the research group, GETALP, the foundation model, Mistral, and its size in number of parameters, seven billions (7B).

References to GETALP models in this work refer to both GETALP-Mistral7B and GETALP-Llama3-8B, an additional variant based on Asclepius-Llama3-8B.

1.5 Reminder

This reminder serves as a roadmap for the reader, emphasizing the logical progression from context and motivation, through methods and experiments, to results, conclusions, and technical appendices.

Chapter 1 introduced the context, motivation, and objectives of the study. It highlighted regulatory requirements, the impact of timely and high-quality discharge summaries on patient outcomes, and the challenges posed by length, usability, and physician workload. Divergent perspectives between hospitalists and PCPs were also discussed, illustrating how differing expectations could influence DS preparation. The chapter concluded by outlining the objectives and contributions of the study, including the textual representation of EHRs, the development of lightweight, modular, instruction-tuned LLMs, and the design of specialized source documents (diary and patient summary).

Chapter 2 reviews the state of the art, providing an overview of LLMs in medicine, distinguishing between biomedical and clinical text, and discussing the role of natural language understanding versus generation in clinical models. It further explores online and open-source models, with a focus on the Asclepius clinical foundation models, and presents the *Discharge-Me!* BioNLP ACL’24 shared task and its results, which serve as a reference for evaluating this work.

Chapter 3 describes the methods employed in this study. It details the design of source documents and their generation, the assembly of model prompts, and

the processing of the MIMIC-IV dataset, including its various modules, diary and patient summary construction, and data splits. The chapter also presents the training of GETALP models using supervised fine-tuning (SFT) and QLoRA-based instruction tuning, along with the evaluation framework, metrics, and the experiments conducted to assess model performance.

Chapter 4 presents the results of the study, including comparisons between GETALP models and their Asclepius baseline, ablation studies on source document content, the use of AI-generated hospital courses in patient summaries, qualitative analysis of model outputs, and the outcomes of the recollection test.

Chapter 5 concludes the thesis with an in-depth discussion on feasibility, source document design, the Mistral7B degeneration problem, the evaluation framework, privacy and recollection concerns, and the relative importance of source document sections.

Finally, Chapter 6 extends this discussion toward future directions, outlining a pathway for the development of an end-to-end DS automation pipeline. It explores the automation of additional DS sections, the use of speech recognition and specialized adapters for clinician–patient interviews, the integration of centralized EHRs, and the processing of non-textual clinical data such as medical images and physiological signals.

The appendices provide further details, including: examples of prompts and source documents (A); MIMIC-IV statistics and considerations regarding the length of source documents (B); technical details on the models, training, and inference, along with an analysis of the carbon footprint (C); and mathematical details underlying the evaluation metrics (D).

Chapter 2

State of the Art

This chapter discusses LLMs and their applications in medicine. In particular, a focus is placed on generative clinical models, as clinical text differs from the polished medical text commonly used for educational purposes. Non-open source solutions such as OpenAI’s GPT models are then examined, which cannot be deployed due to privacy and regulatory concerns. After a review of alternative open-source solutions, One section is dedicated to the clinical base model used as the starting point in this work, Asclepius, which serves as the foundation for the development and fine-tuning of our system. Subsequently, a section introduces the “*Discharge-Me!*” challenge, from which this study originates. Finally, the limitations of the challenge are outlined, and the work is positioned as an alternative approach that aims to overcome such limitations.

2.1 Overview of LLMs in Medicine

LLMs in the medical domain cover both biomedical and clinical texts. While related, these domains differ substantially in style, structure, and data characteristics. The following subsections highlight these differences and discuss the focus of existing clinical models on understanding versus generation tasks.

2.1.1 Biomedical vs Clinical Text

LLMs developed for the biomedical domain cannot be directly applied to clinical text, because the two domains exhibit substantially different characteristics [60]. Biomedical corpora, such as scientific articles, structured databases, and PubMed publications, are typically well-edited, fluent, polished, and free of personal data. **Clinical text**, instead, originates from EHRs and doctors’ notes, often containing errors, undefined abbreviations, sensitive personal data, and a noisier

structure. Table 2.1 summarizes these differences.

Table 2.1: Differences between Clinical and Biomedical Text

Characteristic	Clinical	Biomedical
Errors	yes	no
Personal Data	yes	no
Undefined Abbreviations	yes	no
Edited	no	yes
Fluent	no	yes
Polished	no	yes
Cost	High	Low
Length	Long	Short

As a consequence, biomedical LLMs such as BioGPT [61], BioBART [62], BioMedGPT [63], BioMistral [64], and BioMedLM 2.7B [65] are not suitable starting points for clinical text generation, since they are fine-tuned on a domain that differs fundamentally from the target clinical domain [60].

2.1.2 Natural Language: Understanding vs Generation in Clinical Models

In the field of NLP, tasks can be broadly divided into:

- Natural Language Understanding (NLU): tasks that require comprehension of text, such as name entity recognition (NER) and natural language inference (NLI).
- Natural Language Generation (NLG): tasks that involve producing coherent and contextually appropriate text, such as summarization, report generation, or question answering.

Most clinical LLMs have been developed primarily for NLU tasks. For example, ClinicalBERT [66] and GatorTron [67] are pre-trained on EHRs and discharge summaries to improve representation learning and achieve strong performance on NLU benchmarks. Even models with generative potential, such as **ClinicalT5** [68], are reported and evaluated almost exclusively on NLU tasks.

The open-source **Asclepius** family of models [8], which serves as the foundation for the system developed in this study, is involved in this trend too: Asclepius is trained on eight distinct tasks, five of which are NLU-oriented (e.g., NLI, NER).

2.2 Online LLMs

Several studies highlight the competitiveness of online LLMs in clinical contexts [69, 70, 71, 72, 73]. **OpenAI’s GPT-4** [74] achieves strong performance on multiple NLU in the medical domain, ranking as the leading model on medical language understanding tasks such as MMLU [75], PubMedQA [76], MedMCQA [77] [72]. It has also been evaluated for its ability to **generate realistic clinical scenarios**: thematic analysis of expert reviews confirmed that the cases it produced were clinically accurate, logically consistent, and aligned with established guidelines and nursing competencies [71]. Beyond applications in de-identification [69], GPT-4 achieves high accuracy when **analyzing multilingual clinical notes**, with physicians agreeing with its interpretations in the majority of cases—88% for Spanish, 84% for Italian, and 77% for English [70]. Evidence from Korea further shows that GPT-based systems can generate patient-friendly discharge summaries, with few- and one-shot prompting while maintaining factual accuracy [73]. **GPT-3.5-turbo was used to generate the Asclepius database** [8], producing synthetic clinical notes starting from the PMC-Patient collection of anonymized case reports [78], which serve as the basis for the Asclepius-Mistral7B-v0.3 model.

Med-Gemini demonstrates impressive capabilities in clinical text generation. Saab et al. report that Med-Gemini “surpasses human experts on tasks such as medical text summarization and referral letter generation”, establishing state-of-the-art performance on 14 medical benchmarks spanning text, **multimodal**, and long-context applications where it reaches 91.1% accuracy, slightly above GPT-4’s 90.2% [79]. Yang et al. further show that Med-Gemini can generate clinically acceptable reports from 2D and 3D medical images, with AI reports evaluated as “equivalent or better than the original radiologists’ reports” in a substantial fraction of cases [80].

Despite their strong performance, solutions based on online LLMs are constrained by significant limitations that make them unsuitable for the system. As detailed in 1.2.4, regulatory compliance necessitates an on-premise model. Employing these online models would require transmitting data to an external party and depending on a third-party organization to process patient information; in a clinical environment, the risks of data leakage and potential service disruptions are unacceptable. Moreover, the use of MIMIC-IV is bound by the PhysioNet Credentialed Data Use Agreement [81], which specifically forbids sharing data with third parties or using online platforms such as GPT-4 or Med-Gemini. A final key concern is evaluation fairness: since the training data composition for these models is not disclosed, the possibility that the model has already encountered the test cases cannot be eliminated.

2.3 Open Source Models

This section reviews open-source generative clinical LLMs.

ClinicalGPT [82] is an open-source Chinese LLM. Despite strong results across several medical tasks, the absence of details on training and fine-tuning hinders reproducibility. Its exclusive availability in Chinese further limits applicability in international settings.

SoftTiger [83] is a generative model trained on both general corpora and the MIMIC-IV dataset, primarily aimed at clinical data structuring. While not originally designed for note generation, it has proven highly effective in this task, surpassing ChatGPT-3.5 and performing close to GPT-4. The model is accessible via Hugging Face (HF) in two versions, with 13B and 70B parameters, yet its release raises major concerns. PhysioNet’s policy prohibits distributing models trained on MIMIC-IV to unlicensed users [84], and the lack of clarity on the specific subsets used introduces a risk of data contamination. This may result in a *parrot effect*, where outputs replicate memorized data rather than producing valid inferences, compromising reliability. Given these issues, along with the possibility of removal from public repositories, SoftTiger was excluded from this study.

Clinical Camel [85] builds on LLaMA-2 [86] with QLoRA fine-tuning [10]. It has been extensively evaluated on NLU benchmarks while NLG capabilities are supported only by limited qualitative examples, such as the synthesis of plausible clinical notes. The model also carries the risk of generating misleading or inappropriate content, and a significant performance gap remains compared to frontier models such as GPT-4 and Med-PaLM 2 [87]. Furthermore, Kewon et al. [8] show that Asclepius, even in its 7B version, outperforms Clinical Camel at 13B, despite both use LLaMA-2 as foundation model.

2.3.1 Asclepius

Asclepius is a publicly shareable clinical language model [8], developed to overcome the privacy limitations associated with models trained on real clinical notes.

To this end, a large-scale database of 158k high-quality **synthetic clinical notes** was created from case reports extracted from biomedical literature (PMC Patients [78]). Case reports are detailed, fully anonymized descriptions of individual patients, prepared for academic or educational purposes. These reports were then transformed into DS-style notes using GPT-3.5-turbo, producing a corpus suitable for training a clinical LLM.

Kweon et al. benchmarked Asclepius against multiple models, including general-purpose LLMs (Vicuna [88], Alpaca [89], ChatGPT-3.5-turbo), specialized open-source models (Clinical Camel, Med-Alpaca [90]), and other non-open source solutions (ChatDoctor [91]). Evaluation covered eight task types, of which three

are generative: question answering, **summarization**, and paraphrasing. Scores were assigned by GPT-4 on a discrete five-point scale. Asclepius matches GPT-3.5 in performance and achieves the highest average scores among all other evaluated models. A version trained on real MIMIC-III notes slightly outperforms GPT-3.5-turbo despite being ten times smaller (13B–7B vs 175B).

Because of the combination of strong performance across diverse clinical tasks and an open-source design that enables task-specific adaptation, **Asclepius represents an ideal base model for this study**. In particular, Asclepius-Llama3-8B and Asclepius-Mistral7B were used.

2.4 Discharge-Me! BioNLP ACL’24 Shared Task on Streamlining Discharge Documentation

This work has been inspired by the *Discharge-Me! BioNLP ACL’24 Shared Task on Streamlining Discharge Documentation* [11, 92].

Launched by Stanford University in February 2024, **Discharge-Me!** is a challenge focused on streamlining clinical documentation. Its primary objective was to reduce the time and effort clinicians spend on writing detailed notes. Specifically, the task consisted of generating the hospital course and the discharge instructions. The official dataset of the competition integrates data from MIMIC-IV-ED [93] and MIMIC-IV-NOTE [94]. Precisely, participants were provided with 109,168 visits to the emergency department (ED). Each visit included chief complaints and diagnosis codes documented by the ED, at least one radiology report, and a discharge summary.

The evaluation framework relies on a hidden subset of 250 samples and employs eight complementary metrics, selected to collectively cover three indispensable dimensions of evaluation: lexical overlap, semantic similarity, and medical factual consistency:

- **lexical overlap:** BLEU-4 [37], ROUGE-1, ROUGE-2, and ROUGE-L [38], as well as METEOR [39], assess the degree of lexical similarity between the generated text and the reference. BLEU-4 captures precision over 4-grams, while the ROUGE family emphasizes recall over unigrams, bigrams, and longest common subsequences. METEOR refines overlap-based evaluation by incorporating stemming, synonym matching, and word-order penalties, thereby providing a more linguistically informed measure.
- **semantic similarity:** BERTScore [40] and AlignScore [41] evaluate whether the generated and reference texts convey the same meaning beyond exact wording. BERTScore leverages contextual embeddings from pre-trained language models to align semantically similar tokens, whereas AlignScore introduces

a dedicated alignment model shown to better capture fine-grained semantic fidelity.

- **factual consistency:** MEDCON [42] is a metric introduced within the ACI-BENCH framework to evaluate factual correctness in clinical note generation. It extracts medical concepts from both generated and reference notes using QuickUMLS [95], and computes an F1-score over the resulting clinical concept sets (Anatomy, Drugs, Disorders, Procedures, and others).

For each metric, HC and DI were first evaluated independently—although task-wise scores have not been disclosed. Given a metric, the mean score between HC and DI is then computed. Finally, the overall score of each submission was obtained by averaging over the eight metrics. That score was used to elect the best solution. Furthermore, the top six submissions were evaluated in terms of completeness, correctness, readability, and overall alignment by a panel of clinicians.

2.4.1 Discharge-Me! Results

This section presents the most relevant solutions submitted to the *Discharge-Me!* challenge, ordered by overall score (Table 2.2).

Team	Overall	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
WisPerMed	0.332	0.124	0.453	0.201	0.308	0.438	0.403	0.315	0.411
Yale	0.300	0.106	<u>0.423</u>	0.180	0.284	0.412	0.381	0.265	0.353
aehrc	<u>0.297</u>	0.097	0.414	<u>0.192</u>	<u>0.284</u>	0.383	<u>0.398</u>	0.274	0.332
UF-HOBI	0.286	<u>0.102</u>	0.401	0.174	0.275	<u>0.396</u>	0.289	<u>0.296</u>	<u>0.355</u>

Table 2.2: Results of the *Discharge-Me!* challenge. Best scores in **bold**, second-best scores underlined, R = ROUGE

WisPerMed

The winning team, WisPerMed [96], begun by splitting each discharge summary into its constituent sections using the MIMIC SID algorithm [97, 98]. They then computed the BERTScore of each DS-section against both target sections, in the attempt to rank sections by relevance. On the basis of BERTScores, the discharge summary was re-ordered placing on top the most similar sections, according to the task. Such sorting step ensures that, even after a truncation to 2,000 tokens, important clinical notes are yet at a model disposal.

WisPerMed fine-tuned multiple models, training a separate model for each task (HC and DI). Fine-tuning employed the LoRA [99] technique, which specializes

small adapter modules while keeping the base model frozen. Several LLMs were fine-tuned and prompted, and a dynamic expert selection (DES) strategy was applied to select the most suitable output. DES first ranks models according to their overall score, then, for each discharge summary, it chooses the generated section whose word count fell within a target range of 100–180 words, with fallbacks down to 70 words if necessary. Considering the models’ performances individually—without DES—**Asclepius-Mistral7B** achieved the highest overall score.

HarmonAI Lab at Yale

Socrates et al. [100] first extracted all relevant structured information for each patient, including demographics, ED diagnoses, procedures, inpatient medications, and lab results. Using these extracted sections, individual SOAP notes were generated by GPT-3.5 Turbo in a zero-shot setting.

As WisPerMed, Yale fine-tuned **one model per task**. Two families of models were employed: BioBART-Large [62] and Clinical-T5-Large [68], while also leveraging Azure OpenAI GPT-3.5 Turbo in a zero-shot setting with human-based abuse monitoring switched off, in compliance with the MIMIC-IV DUA [81]. The team additionally employed constrained generation, injecting external knowledge about the preferred style of the HC to account for formatting variations across physicians and discharge wards, as observed through manual analysis of clinical notes.

aeHRC

Liu et al. [101] developed their best model using PRIMERA [102], an encoder-decoder language model capable of handling extended input contexts and generating longer outputs. The authors observed that including **radiology reports could lead to hallucinations** while providing negligible performance improvements.

UF-HOBI

The UF-HOBI team [103] proposed a method that combines medical concepts detected via NER with selected discharge summary sections. Discharge summaries were first segmented into sections (e.g., History of Present Illness (HPI), Discharge Conditions, Physical Exams) using a **rule-based** algorithm. Sections were then divided into two groups: those strictly relevant to the output, and those useful but non-essential. Concepts from the latter group were extracted with GatorTron [67] via NER and then provided to the generative model, GatorTronGPT [104].

2.4.2 Limitation of the *Discharge-Me!* Solutions

Several top-performing solutions highlight the mismatch between competition-oriented optimization and real-world applicability. WisPerMed, for example, trains multiple models and selects the final output based on length criteria. While this improves leaderboard metrics, running a dozen models in parallel is impractical for hospitals with limited resources and **does not reflect a deployable workflow**. Similarly, Yale’s constrained generation approach enforces stylistic tokens in the output based on metadata, as the department or the author of the ground truth summary are. This encourages the model to replicate the style of discharge reports from the Beth Israel Deaconess Medical Center, which boosts BLEU, ROUGE, and METEOR scores within the challenge, but limits flexibility and broader applicability. A real-world system should instead adapt to the style indicated by physicians.

While the *Discharge-Me!* challenge represents an important step toward evaluating models for clinical documentation, it falls short of its stated goal of streamlining the discharge process. The competition relies on discharge summaries as the primary input to the model, which means that physicians still need to manually compose detailed clinical notes before any automation can take place. This contradicts the intended objective: a system that truly reduces the time and effort required to prepare discharge documentation should rely on more direct and raw data sources, such as auto-filled medication records at the time of administration or lightweight structured forms that can be completed in seconds.

By relying solely on discharge summaries as the only source of information, current approaches fail to address the underlying goal of reducing physicians’ administrative workload.

2.4.3 Deploying to Overcome Limitations

In this work, a different usage of data is adopted, in order to better reflect real-world practice while retaining *Discharge-Me!* objectives and evaluation framework. Specifically, more modules from MIMIC-IV are employed, while third DSES-sections are only used to fill-in information gaps. For example, details such as conditions at discharge or physical examination are only available in MIMIC-IV’s discharge summaries and therefore they must be retrieved from there. Other elements can be directly sourced from the corresponding tables (e.g. microbiology results from `microbiologyevents.csv`).

Since the **hidden competition test samples have not been made public**, a similar set is constructed by randomly sampling 250 admissions while carefully avoiding data contamination. This ensures a consistent evaluation protocol and allows to benchmarking GETALP models against *Discharge-Me!* solutions.

Chapter 3

Methods

This chapter provides a comprehensive overview of the technical procedures and experimental setup employed in this study, to allow reproducibility of the results. It begins by introducing the **source documents (SDs)**, task-specific textual representations of clinical data designed to standardize and consolidate information from heterogeneous EHRs. Two types are defined: the **diary**, tailored for HCs, and the **patient summary (PS)**, aimed at DI generation.

Subsequently, the chapter details the **SD generation pipeline**, made of two components: the LogBuilder and the DiaryMaker. Finally, the **Prompt Assembler**, which organizes the textualized data into model-ready input prompts, is explained.

The dataset utilized for model training and evaluation, **MIMIC-IV**, is described with a focus on relevant modules (HOSP, NOTE, ED) and the construction of the source documents from it. Data splits, source document lengths, and preprocessing steps are also covered, with detailed statistics provided in Appendix B.

Next, the **training procedures** are outlined, including both supervised fine tuning (SFT) and QLoRA for efficient, quantization aware model adaptation. Training hyperparameters, memory usage, and learning curves are reported in Appendix C.

The **evaluation framework** section introduces the metrics used to assess model performance, detailing per-task evaluation methods and the overall scoring system. The mathematical formulation of all metrics is provided in Appendix D.

Finally, the chapter presents the conducted **experiments**, covering model evaluation, ablation studies on the diary content, the integration of GETALP-generated hospital courses into patient summaries, qualitative analysis of outputs, and recollection test.

3.1 Source Documents

LLMs are pre-trained on vast corpora of free text. EHRs, however, store much of their content in structured—often tabular—formats, which cannot be directly consumed by an LLM. To bridge this gap, EHRs are transformed into textual documents, referred to as **source documents** since they constitute the source of information for GETALP models. It is worth noting that SDs are not conventional medical documents: they are not intended for human use. Rather, they are engineered for LLMs, with efficiency in both content and space as the primary goals.

Two distinct SDs are introduced. The first, called the **diary**, is designed to capture the complete sequence of events during a hospitalization, thereby supporting the generation of a coherent hospital course. The second, named the **patient summary**, condenses and organizes only the key elements necessary to produce safe and informative discharge instructions.

The need for two separate documents arises from the nature of their respective objectives. Writing a hospital course requires the model to track the chronological order of all procedures and treatments, while producing discharge instructions demands a concise overview of diagnoses, conditions, and follow-up care. Accordingly, the diary emphasizes sequentiality and completeness, whereas the patient summary focuses on conciseness and relevance to post-discharge care.

Although the database used in this study is MIMIC-IV, the procedure is not tied to it: any EHR system can be reorganized into diaries and patient summaries, provided that its underlying data schema is known and programmatic access is available. The approach, along with the SD layouts themselves, is therefore **generalizable beyond MIMIC-IV**.

In the following subsections, the design of the diary (3.1.1) and the patient summary (3.1.2) is detailed, explaining how each layout is constructed from EHRs and the rationale behind the design choices.

3.1.1 Diary

A hospital course is a concise summary of a patient’s hospitalization, narrating the sequence of events from admission to discharge. Producing an accurate summary requires a clinician to have a complete understanding of what occurred throughout the hospital stay. In the same way, an AI assistant generating a hospital course must be provided with the same information in the order it happened, as the summary inherently reflects the temporal progression of the hospitalization; without this structured input, the model cannot generate a coherent and clinically meaningful account.

The diary is the input document specifically designed to facilitate the generation

of high quality hospital courses using a large language model. It is constructed to reflect two essential properties: **sequentiality** and **completeness**. Sequentiality ensures that the history of a hospitalization is presented in chronological order, while completeness guarantees that all events relevant to the clinical case are captured.

A type of document known to exhibit these properties is a log file. A log file is a computer-generated record that lists events and operations chronologically within a system, providing a detailed, time-stamped (YYYY-MM-DD-hh:mm:ss) account of what has occurred. **The diary is designed to mirror the structure of log files.**

Hospital stays are characterized by a large number of clinical events, which in turn can lead to lengthy patient diaries. For instance, in MIMIC-IV a typical patient admission is associated with an average of 80 events, with substantial variability. Each hospital stay also includes an average of four radiology notes, with approximately 170 words each, and other notes such as the history of present illness (around 300 words). This can easily result in a diary whose length exceeds the token limits of clinical AI models, requiring a careful design of the document layout that balances completeness and compactness.

Although the length of a diary varies with the number of extractable clinical events in an EHR system, its typical word and event counts can be estimated from the MIMIC-IV database, as shown in Appendix B.

Diary Layout

As anticipated, the diary is inspired by log files, which typically record events as long timestamp–event pairs (e.g., 2025-09-08-13:10:00 Prescription Omeprazole 40mg Injection IV DAILY). However, to avoid excessive verbosity and reduce token usage, information is reorganized into a more compact structure based on pages and logs.

Each page groups all events that occurred on a given day. A page begins with a header line of the form -- YYYY-MM-DD -- and ends when the header of the following page is encountered. Pages are created only for days on which at least one event occurs, so there are no empty pages. Starting from the admission date, consecutive pages are stacked in chronological order, producing a longitudinal record that spans the entire hospitalization until discharge.

Within each page, events are recorded as *logs* (also called *entries* or *lines*). A log follows the triplet format [time-action] message, where the **time** indicates the hour and minute of occurrence (hh:mm), the **action** specifies the category of activity (e.g., Consults, Prescriptions), and the **message** conveys the details.

When multiple messages correspond to the same time-action pair, they are aggregated under a single log. As a result, each time-action pair within a page is

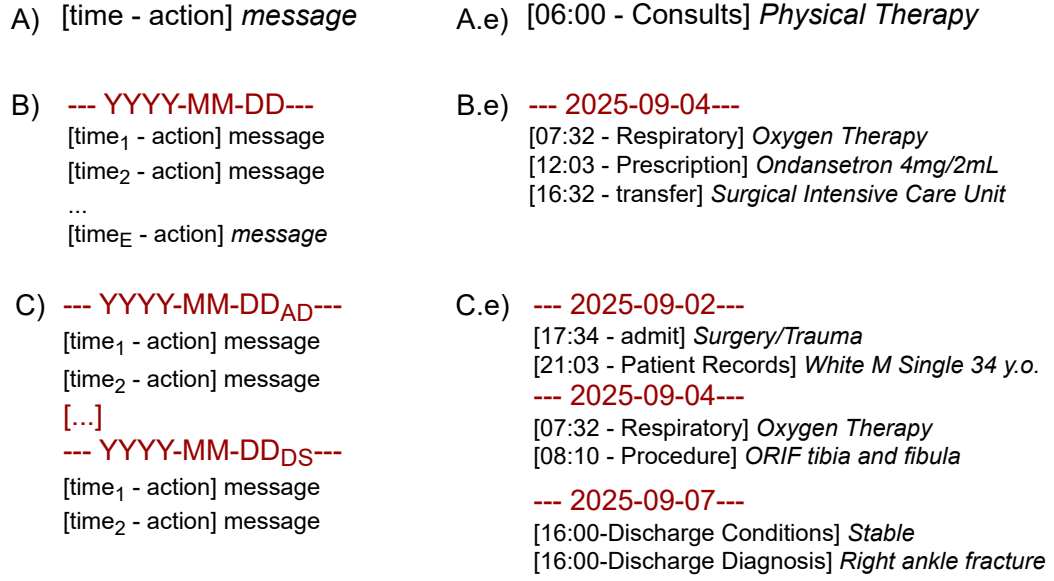


Figure 3.1: Diary. The diagram shows the diary (C), which is composed of daily pages (B), each containing individual logs (A). Examples of log entries (.e) are taken from MIMIC-IV.

unique. For example, if both Omeprazole and Enoxaparin are prescribed at 13:10 on a given day (e.g. 2025-09-08), only a single log is created, with the message listing both prescriptions:

```
--- 2025-09-08 ---
[13:10 - Prescriptions]
- Omeprazole 40mg Injection IV DAILY
- Enoxaparin 40mg Subcutaneous DAILY
```

This organization preserves sequentiality while significantly reducing redundancy. By replacing long timestamps with compact daily headers, grouping events that share the same **time-action**, and omitting empty days, a diary reports the complete trajectory of care in a concise form. The result is a coherent, chronological timeline of patient care that remains suitable for large language model input without exceeding context length.

A MIMIC-IV's diary can be found in Appendix A.2.

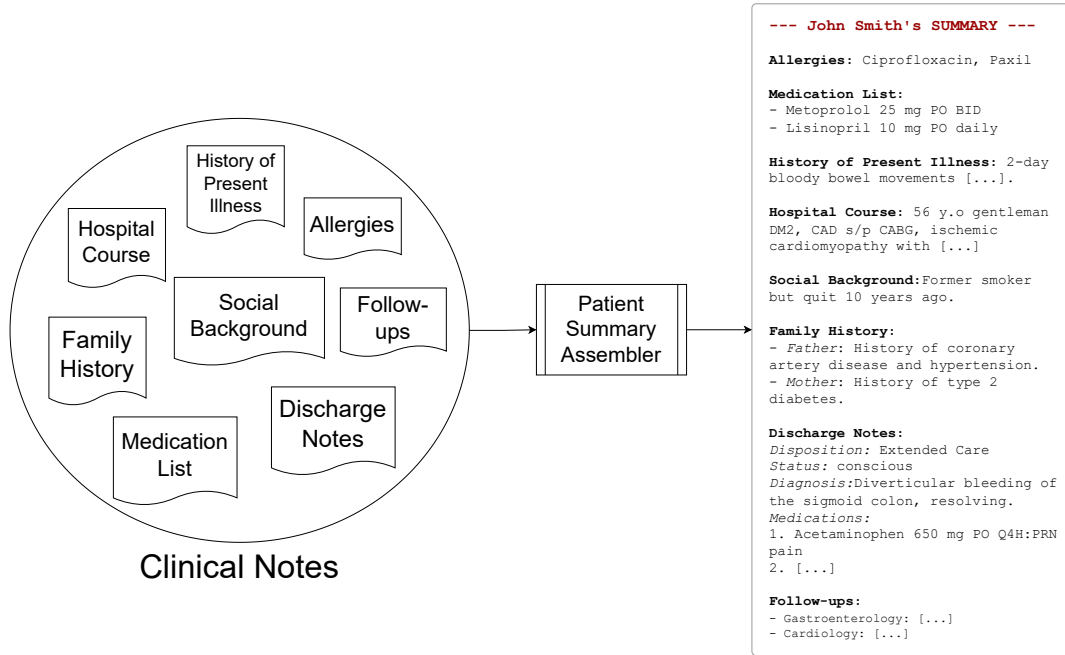


Figure 3.2: Patient Summary. The patient summary is generated from clinical notes and serves as input for an AI assistant tasked with generating discharge instructions for a given patient. The document names shown are illustrative and do not correspond to any specific EHR system.

3.1.2 Patient Summary

Discharge instructions present a concise summary of the hospitalization in lay language, covering the reasons for admission, major procedures performed, instructions regarding *dos and don'ts*, any changes to the patient's usual medications, and follow-up recommendations. The document is designed to be accessible and easily understood by patients, supporting comprehension rather than clinical decision-making.

The patient summary is the source document designed to prompt GETALP models to generate discharge instructions. It consists of clinical notes that summarize the key aspects of a patient's hospital stay, highlighting the most relevant information for post-discharge care. A patient summary includes patient personal information, the history of present illness (HPI), the hospital course, diagnoses, conditions at discharge, and follow-up medications.

Among the patient's personal information, family history and social context provide essential background for contextualizing the discharge plan. Family medical

history may indicate hereditary risks or conditions that could influence post-discharge care, such as cardiovascular diseases or diabetes. Similarly, the social context offers insight into the patient’s living conditions, support network, and potential barriers to adherence (e.g., limited mobility, social isolation, or economic constraints). By exposing these details to the system, the generated discharge instructions can better reflect the patient’s real-world circumstances.

The HPI is equally crucial, as it encapsulates the clinical reasoning behind the hospitalization. It outlines the evolution of symptoms, relevant prior treatments, and the events that led to the current admission. Including the HPI in the patient summary enables the model to understand the causal chain connecting the patient’s initial presentation to their hospital course and subsequent recovery. This temporal and etiological context helps the model to write discharge instructions consistent with the underlying pathology and the therapeutic interventions applied during hospitalization.

Patient Summary Layout

The patient summary organizes clinical notes in a structured order to support discharge instructions generation. Notes follow this sequence:

- **Patient Personal Information:** allergies, medication reconciliation, family medical history, social context;
- **History of Present Illness;**
- **Hospital Course;**
- **Discharge Disposition, Diagnoses, Conditions, Medications.**

In case an information is missing, it is simply ignored.

This layout ensures all relevant information is captured, giving GETALP models the clinical context for accurate and reliable discharge instructions generation.

A MIMIC-IV’s patient summary can be found in Appendix A.2.

3.2 Source Document Generator

This part of the report illustrates the procedure to convert structured tables into source documents.

Creating a patient summary simply involves assembling the relevant clinical notes into a textual document. In contrast, creating a diary requires integrating multiple types of tables and notes, necessitating a careful engineering process to ensure completeness and sequentiality.

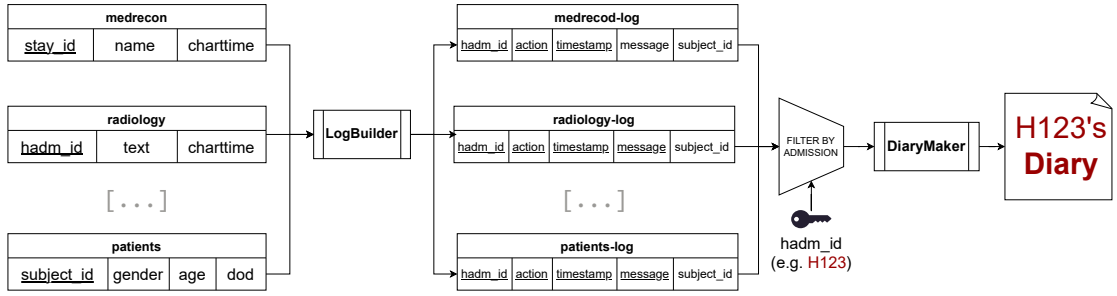


Figure 3.3: Diary generation pipeline. Each table is processed by the LogBuilder, which standardizes its schema into a fixed format containing subject and admission identifiers, timestamp, action, and message. Given an `hadm_id`, the relevant entries are selected from the log tables and assembled by the DiaryMaker in chronological order to generate the diary. All tables shown are derived from MIMIC-IV

For this reason, the **Source Document Generator for the patient summary is limited to assembling clinical notes in a pre-defined sequence**, determined by the note types available in the EHR system in object. Given the simplicity of this process, its technical implementation is not discussed further. The specific methodology for constructing the patient summary in MIMIC-IV is detailed in Section 3.4.6.

The creation of a diary from EHRs is a three-stage process. The first stage involves a manual analysis of the database to identify relevant clinical data, its temporal features, and the necessary joins and transformations. This analysis is exclusively used to design the **LogBuilder**, an EHR system-specific component that converts raw, heterogeneous tables into tables with a unified schema. The final stage is handled by the **DiaryMaker**, an algorithm that assembles this standardized log tables into a diary. A schematic representation of this process is provided in Figure 3.3.

3.2.1 LogBuilder

The LogBuilder is the algorithm that converts EHR tables into a unified log format. It is the only component of the system depicted in Figure 1.5 that must be **re-engineered** if the underlying health database changes. Its design is based on a preliminary database analysis that involves three key steps:

- identifying the temporal feature to track the order of events;
- joining complementary tables together;
- selecting relevant features.

Based on the insights gained from this manual analysis, a LogBuilder is then designed and tailored to the specific EHR system. Its goal is to **reshape the data into a standardized format** containing only five key fields: `hadm_id`, `subject_id`, `timestamp`, `action`, and `message`. This standardized structure **allows the DiaryMaker to operate agnostically with respect to the original schema**.

Defining the content of the `action` and `message` fields is the core of the EHR-to-Diary transformation. The `action` may represent the name of the source table itself or a specific pattern within a subset of data, while the `message` is generated by reorganizing the table’s values into a human-readable string. Notably, every field is assigned to one of these two categories to compress the conveyed information as much as possible. This entire data mapping strategy stems directly from the initial manual analysis and represents the core logic of the LogBuilder.

The LogBuilder is composed of several submodules, each designed to process a specific part of the EHR content. The design of the LogBuilder for MIMIC-IV is reported in Section 3.4.5.

From this point onward, all processing steps remain identical regardless of the underlying EHR system.

3.2.2 DiaryMaker

The **DiaryMaker is the algorithm responsible for creating the diary from the standardized log tables** produced by the LogBuilder.

Given a hospitalization identifier (`hadm_id`), the DiaryMaker collects all entries related to that admission from the log tables. Once the events are gathered, the algorithm splits the `timestamp` field into two parts: the date (`YYYY-MM-DD`) and the time (`hh:mm:ss`). Events are then grouped by date, creating one page for each hospital day.

Within each day, multiple records may share the same `time-action` pair. In such cases, all messages associated with that pair are aggregated into a single list, as described in Section 3.1.1. This condensation step ensures that **each time-action pair uniquely identifies an event**. An event is therefore defined by the triplet:

[time-action] message

Each page is filled with its corresponding events, represented as such triplets. Pages are then ordered chronologically, from admission to discharge, and concatenated to form the diary.

3.3 Prompt Assembler

The source document provides the clinical context required for each task but, on its own, it is insufficient for the model to produce accurate outputs. To guide the model, the SD is combined with additional task-specific information that helps the model interpret its objective, forming the **prompt**.

Although there is no universally accepted nomenclature for the parts of a prompt, in this framework it is divided into three main components, each serving a distinct purpose: the system message, the clinical context, and the instructions.

The first component, the **system message**, provides the model with information about the task and the context document that will serve as its main reference. It also defines the model’s role and establishes general guidance on how the generation should be approached.

The second component, the **context**, constitutes the core informational content and varies for each individual case. While the system message and instructions remain fixed for a given task type, the context is specific to the individual case under analysis, depending on the user’s request and the available clinical data.

The third component, the **instructions**, specifies in detail what the model is expected to produce and how. They define the structure, content, and style of the output, guiding the model to generate information that is accurate, coherent, and suitable for the intended audience.

In this work, the system message provides a high-level description of the task and its associated context; the context is represented by the source document, and the instructions specify the required output format, the organization of the information, and the desired writing style, ensuring that the result is appropriate for clinical use.

In summary, the prompt integrates the clinical context with precise task guidance, enabling instruction-tuned models to produce high-quality, task-specific outputs. The **Prompt Assembler** is the algorithm responsible for constructing these prompts, combining the system message, context, and instructions for each task. Examples of prompts used to feed GETALP models are provided in Appendix A.1.

3.4 MIMIC-IV

This section introduces the EHRs database used in this study: the **Medical Information Mart for Intensive Care** in its **fourth** edition (**MIMIC-IV**) [9]. MIMIC is a large, freely available database containing de-identified health-related data from patients admitted to the critical care units of the Beth Israel Deaconess Medical Center (BIDMC). Although **anonymized**, the datasets include detailed information about patient care. Researchers wishing to use the database must

become credentialed users on PhysioNet [84] and sign the Data Use Agreement (DUA) [81].

The database is organized into six modules: **HOSP**, **ED**, **NOTE**, ICU, CXR, and ECG, each capturing data from different stages of care. Most data are stored as comma-separated values (*.csv*) files, while some modules, such as CXR and ECG, also contain images and waveform data. Each module contains multiple tables, and Figure 3.4 shows only the tables utilized by the system, with the corresponding modules represented by an open folder icon.

The content, organization, usage, and data pre-processing of each module are described in the initial sections. This is followed by two sections detailing the source documents, namely the MIMIC-IV diary and the MIMIC-IV patient summary, and concludes with the subdivision of the dataset into training and testing splits.

3.4.1 *ED* Module

Data from the emergency department are recorded in six tables: *edstays*, *medrecon*, *triage*, *diagnosis*, *pyxis*, *vitalsigns*. Each ED stay is identified by a unique numerical code, *stay_id*. Multiple *stay_ids* can correspond to a single *hadm_id*, since a patient may visit the ED several times before being admitted. Occasionally, patients may leave the ED without being admitted or may be unable to afford hospitalization.

The *edstays* table plays a central role in this study, as it contains all primary keys: *subject_id*, *hadm_id*, and *stay_id*. During data pre-processing, when merging different tables, some keys may be missing; in such cases, *edstays* is used to retrieve them.

Medication Reconciliation—the process of comparing a patient’s current medication orders with those they had been taking—is captured in *medrecon*.

For training GETALP models, only two tables of this module were used: *edstays* and *medrecon*.

3.4.2 *HOSP* Module

The HOSP module contains most of data captured by the BIDMC electronic health records system. These measurements are primarily **recorded during hospital stays**, though some tables also include information from outside the hospital (e.g., outpatient laboratory tests in *labevents*). The available information includes patient and admission details (*patients*, *admissions*, *transfers*), laboratory measurements (*labevents*), microbiology cultures (*microbiologyevents*), provider orders (*poe*, *poe_detail*), medication administration (*emar*), prescriptions (*prescriptions*, *pharmacy*), billing information (*diagnoses_icd*, *procedures_icd*, *hccpcsevents*, *drgcodes*), and hospital service information (*services*).

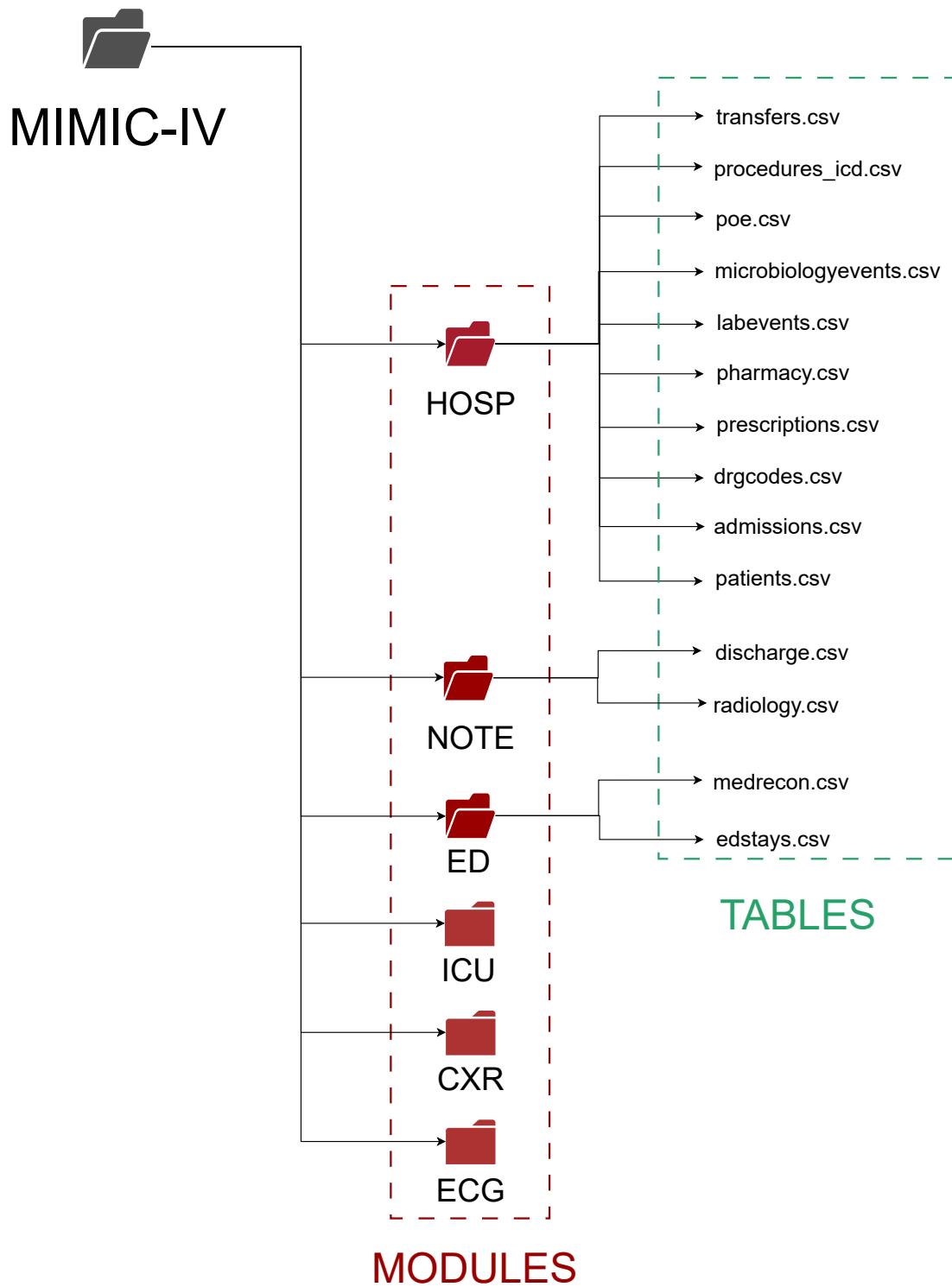


Figure 3.4: MIMIC-IV structure. Comprehensive documentation is available at mimic.mit.edu.

Each hospitalization is identified by a unique `hadm_id`, a numerical code assigned to an admitted patient. The `hadm_id` serves as the primary key for this module and should not be confused with the `subject_id`, which identifies patients at BIDMC. A single patient (`subject_id`) may have multiple admissions, each with a distinct `hadm_id`. Each record has a **temporal attribute** which can be used to reconstruct the chronological order of events during a hospital stay.

Not all HOSP’s tables were used to train and validate GETALP models. Some contain redundant information, such as *emar*, which adds negligible details beyond *pharmacy* and *prescriptions*. Some other tables contain secondary or ancillary information, such as billing or miscellaneous records, which could introduce noise into the model. For instance, the *diagnoses* table includes secondary diagnoses and chronic conditions, whereas *drgcodes* focuses on the primary reason for care. Therefore, *drgcodes* were used, as it was considered that less critical illnesses and chronic conditions can still be captured indirectly through other data, such as medications and procedures. **The excluded tables are: *omr*, *emr*, *hcpcsevents*, *diagnoses*, and *provider*.** Table 3.1 summarizes the HOSP tables included in this study.

3.4.3 NOTE Module

MIMIC-IV-NOTE [94] contains radiology reports and discharge summaries. The primary key for both tables is the `hadm_id`.

Discharge Summaries

Discharge summaries are essential clinical documents that provide a comprehensive account of a patient’s hospitalization. They serve as handover tools, ensuring continuity of care from the hospital to subsequent healthcare providers, including primary care physicians, specialists, or long-term care facilities. At the BIDMC, they are generally organized in a near standardized format, with distinct sections documenting various aspects of the patient’s clinical course and care. MIMIC-IV discharge summary sections are reported in Table 3.2.

To develop GETALP models, each discharge summaries is divided into its component sections through a structure-based algorithm designed in this study, called **DShunker**. This allows retrieving the target sections (HC, DI) and extracting other clinically valuable information that cannot be obtained from structured HOSP or ED data. In MIMIC-IV, the section corresponding to the patient’s hospital course is officially named Brief Hospital Course (BHC).

Some sections, such as History of Present Illness, Allergies, Discharge Condition, and Discharge Medications, are embedded in the discharge summaries, therefore

Table	Features	Exceptions
transfers	care unit; event type	entries with unknown care unit
procedures_icd	procedure description	none
poe (Provider Order Entry)	order type; order subtype; field name; field value	duplicates (e.g., <i>ADT</i> , <i>medications</i> , <i>labevents</i>); routine and non-informative orders (e.g., nutrition, blood bank, nursing, telemetry, IV access)
microbiologyevents	specimen type; organism name; comments	samples with no growth; missing organism names replaced by comments
labevents	test name; result value; unit; comments	none
prescriptions, pharmacy	drug name; product strength; administration route; frequency; dispensing schedule; sliding scale	prescriptions labeled as <i>BASE</i>
drgcodes	diagnosis	none
admissions	admission type, time and location; insurance; hospital expiration flag; discharge time and location; death time	none
patients	race; gender, marital status; age; language	none

Table 3.1: Overview of HOSP tables. *Features* lists the key fields extracted from each table. *Exceptions* specifies the criteria used to remove records from the original datasets, such as missing values, duplicates, or entries considered non-informative.

they cannot be retrieved from elsewhere in MIMIC-IV. Other sections, like Social History and Family History, cannot be reconstructed because **MIMIC-IV’s de-identification process breaks any link** with patient relatives or social background. Additionally, sections such as Past Medical History could in principle be partially derived by aggregating previous hospitalizations through the `subject_id`, but the result would be incomplete: these sections often include details collected

Section	Content	Included
Demographics	Name, language, race, gender	No
Service	First ward responsible for the patient	No
Allergies	Drug, food, or environmental allergies (reaction/severity)	Yes
Attending	Primary physician overseeing care	No
Chief Complaint	Reason for admission	Yes
Major Surgical or Invasive Procedures	Operations or significant interventions during hospitalization	Yes
History of Present Illness	Detailed narrative of symptom onset, progression, and severity	Yes
Past Medical History	Prior diagnoses, chronic conditions, previous hospitalizations	Yes
Social History	Lifestyle factors, habits, occupation	Yes
Family History	Familial health patterns	Yes
Physical Exams	Clinical findings at admission and at discharge	Yes
Pertinent Results	Key laboratory, imaging, and diagnostic findings	No
Medications on Admission	Drugs taken prior to hospitalization	No
Discharge Medications	Prescribed drugs at discharge, including dosing	Yes
Discharge Disposition	Post-hospital care setting	No
Discharge Diagnosis	Primary and secondary diagnoses at discharge	Yes
Discharge Condition	Clinical status at discharge	Yes

Table 3.2: Overview of discharge summary sections in MIMIC-IV-NOTE, including content and usage. The last column (*Included*) indicates whether the section was included in the diary or not.

during triage or at hospital admission, sometimes under **dictation by patients or relatives**, and may refer to events outside the BIDMC.

For these reasons, **all sections that cannot be reliably deduced from structured data are sourced from the discharge summaries**. This choice aims to design a system that can be realistically applied in clinical practice. If clinicians were required to manually recompile information already present in the

EHR database, the purpose of automating discharge documentation would be undermined. Instead, clinical staff should only fill in the necessary fields through the EHR system and manually complete sections when truly unavoidable—precisely in cases where the information is only known to the patient, such as the history of symptom onset or the description of a trauma.

The **DSChunker algorithm** processes each discharge summary line by line and searches for section headlines, which correspond to the section names reported in Table 3.2. Headlines are detected using regular expressions, which ensure robustness to small formatting variations such as punctuation, capitalization, or extra spaces. Once a headline is found, all subsequent text is assigned to that section until the next headline is encountered. The algorithm is resilient: if a section is missing or if sections appear in an unusual order, extraction proceeds without issue. Sections require no post-processing, except for Physical Exam, which is further divided into admission and discharge subparts by detecting wake words such as “Admission” or “Discharge.” If no wake word is found, the content is assumed to belong to the admission phase.

The DSChunker **organizes** the extracted information into two tables. The first, `ds_chunk`, collects all sections of each discharge summary, with one row per summary referenced by `hadm_id` and one column per section, as shown in Figure 3.5. Admission and discharge times are retrieved from the `admissions` table of MIMIC-IV-HOSP. The second table, `ground_truths`, stores the target sections (HC and DI) separately for the training phase.

Table 3.2 reports all sections of MIMIC-IV discharge summaries, indicating whether they are included in this study and whether they are labeled as admission- or discharge-related. Sections such as Medications or Pertinent Results are excluded because their content can be obtained directly from prescriptions, pharmacy, laboratory, or microbiology data. By contrast, sections that cannot be reconstructed from structured sources are retained, together with the target sections HC and DI.

Radiology

Radiology reports are a formal, written interpretation by a radiologist of findings from a medical imaging test, such as an X-ray, MRI, CT scan. The style of a radiology note may vary according to different factors, such as the type of exam. Most of documents are split in five different sections: Examination, Indication, Technique, Comparison, Impression. However, style and content of such notes is **less standard** if compared with discharge summaries. Hence, the radiology note is not split in subsections when included in the patient’s diary. Unlike discharge summary, radiology notes come with a precise temporal indication, which allows these notes to be positioned at the correct point in the patient’s history. According to MIMIC-IV official documentation, such variable reports the time at which the

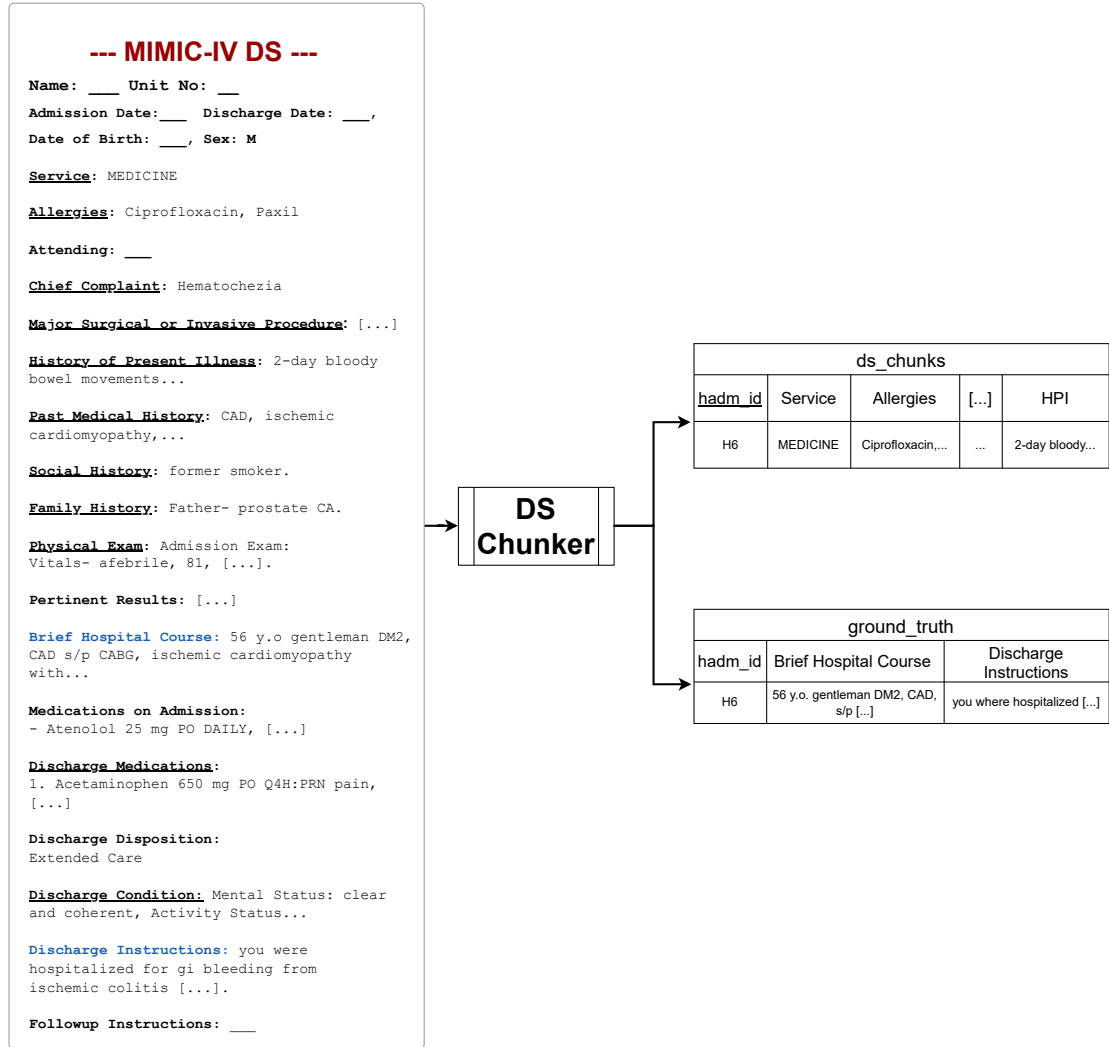


Figure 3.5: MIMIC-IV discharge summary chunking process. A MIMIC-IV DS is shown with its most common layout, showing the order and names of sections. Section names are in bold. Sections underlined are used for training and testing the GETALP models. The two target sections, the Brief Hospital Course and the Discharge Instructions, are written in light blue. The DSChunker algorithm behaves as detailed in Section 3.4.3.

note was charted, that is usually the most relevant time for interpreting the content of the note.

3.4.4 Other Modules: *ICU, CXR, ECG*

The ICU, CXR, and ECG modules correspond to data from the Intensive Care Unit, Chest X-Ray, and Electrocardiogram, respectively. While the ICU module organizes information in tables, CXR and ECG also include images and waveform data. **None of these modules were used in the present study.**

3.4.5 MIMIC-IV Diary

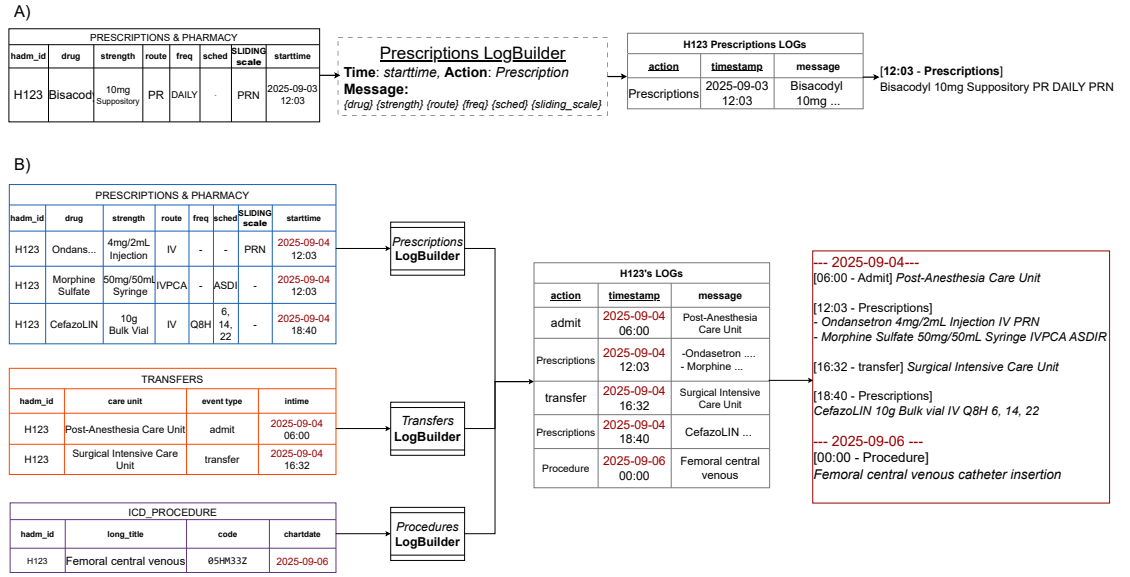


Figure 3.6: MIMIC-IV-to-Diary transformation process exemplified using three tables. Part A shows how a row from *prescriptions & pharmacy* tables is converted by the corresponding LogBuilder-submodule into a log entry. The Prescription LogBuilder extracts **starttime** as the temporal field, fixes the action to the word “Prescription”, and encodes the remaining attributes (**drug**, **strength**, **route**, **frequency**, **schedule**, **sliding scale**) into a concise textual message. Some submodules also adjust temporal information: as shown in B, the Procedures LogBuilder places events at midnight ensuring they appear before other same-day events, since *icd_procedures* records only the date of a procedure (**chartdate**). Part B shows how the DiaryMaker consumes the collection of log tables, merging rows with identical timestamp-action pairs into a single entry (e.g., combining ondansetron and morphine sulfate), ordering entries by **timestamp**, and grouping them into daily pages (e.g., 2025-09-04, 2025-09-06), which are concatenated sequentially to form the complete diary (red rectangle).

This part illustrates how MIMIC-IV tables are converted into log-tables through

the LogBuilder, and how these logs are subsequently assembled into a diary. Once the LogBuilder has standardized the data into a unified log format, the construction of the diary proceeds identically for any EHR system. In other words, **nothing specific to MIMIC-IV is required beyond the general SD generation process** outlined in Section 3.2. Figure 3.6 shows an example of the MIMIC-IV-to-Diary transformation process detailed in this section of the report.

MIMIC-IV LogBuilder

The LogBuilder converts MIMIC-IV structured tables into log-tables suitable for constructing the diary. Besides patient and encounter identifiers (`subject_id`, `hadm_id`), each log entry contains the following fields:

- **timestamp**: date (YYYY-MM-DD) and time (hh:mm:ss) associated with the event
- **action**: a label representing the macro-type of event
- **message**: a human-readable string summarizing the event

The algorithm begins by loading each relevant MIMIC-IV table, retaining only entries with valid `subject_id` and `hadm_id`.

Next, the `action` and `message` fields are generated. In the default case, the `action` is derived from the table name or event type, while the `message` is constructed as a human-readable string from the relevant table fields. Some tables require more complex handling.

For *labevents*, lab results are grouped by patient, admission, fluid, and test label. Only changes in status produce log entries: abnormal results generate an entry labeled “LAB-Abnormal”, and recovery to normal generates an entry labeled “LAB-Resolved”. The `message` summarizes the test name, result, unit, and any comments.

For *microbiologyevents*, the set of detected organisms is tracked for each patient, admission, and specimen type. Log entries are created only when the set of organisms changes. The `message` lists the specimen and the identified organisms, while the `action` is set to the word “MICROBIOLOGY”.

For the *admissions* table, logs are split into two entries per hospitalization: *admit* and *disch*. During admission, the `hospital_expire_flag` is converted to “YES”/“NO”; if the patient died during the hospitalization, “YES” is recorded along with the `deathtime`, otherwise only “NO” is recorded in the message. The `action` is set to “Admission” for admission events, whereas “Discharge” is used for discharge events.

Timestamps are assigned from the most relevant datetime column in each table (e.g., `charttime`, `starttime`, `admittime`, `disctime`). **When only the date is**

available, as in the case of ICD procedures (*icd_procedures*), **the timestamp is set to midnight of that date**. This ensures that the event is placed at the beginning of the day, preserving cause-effect relationships with other events that have more precise timestamps.

Finally, entries with empty **message** fields are removed, producing the final log-table. This log-table is then ready for downstream processing in the diary generation pipeline.

Tables 3.3 and 3.4 summarize, for each table, how **timestamp**, **action**, and **message** are constructed.

Table	Timestamp	Action
medrecon	charttime	Medrecon
transfers	intime	<i>eventtype</i>
procedures_icd	chartdate	Procedure
poe	ordertime	<i>order_type</i>
microbiologyevents	charttime	Microbiology
labevents	charttime	LAB-Abnormal / LAB-Resolved
prescriptions	starttime	Prescription
drgcodes	disctime	DRG Code
admission (admit)	admittime	Admission
admission (disch)	disctime	Discharge
patient_records	admittime	Patient Records
radiology	charttime	Radiology
ds_chunks	admittime or disctime	<i>section_name</i>

Table 3.3: MIMIC-IV log-tables: timestamp and action for each table. One log entry for each section within *ds_chunks* is created.

MIMIC-IV DiaryMaker

The MIMIC-IV DiaryMaker **operates as for any other EHR system**.

For each hospitalization (**hadm_id**), events with the same **timestamp** and **action** are merged into a single entry, with messages listed as a dash-separated list. Events are then grouped by calendar date (YYYY-MM-DD), one page is created for each date, and the page is filled with entries in the format **[time-action] message**, where the **message** may consist of multiple items.

3.4.6 MIMIC-IV Patient Summary

In the MIMIC-IV dataset, as of 2025, only two types of clinical notes are available: the discharge summaries and radiology notes. To create the patient summary, **clinical notes are simulated by splitting the discharge summary into its**

Table	Message
medrecon	<code>name</code>
transfers	<code>careunit</code>
procedures_icd	<code>long_title</code>
poe	<code>order_subtype field_name field_value</code>
microbiologyevents	<code>spec_type_desc findings: current_orgs</code>
labevents	<code>label value valueuom comments</code>
prescriptions	<code>drug prod_strength route freq sched sliding_scale</code>
drgcodes	<code>description</code>
admission (admit)	<code>type, from loc, insurance: ins. Died in Hospital?: exp? dtime</code>
admission (disch)	<code>discharge_location</code>
patient_records	<code>race gender marital_status anchor_age y.o. language speaker.</code>
radiology	<code>text</code>
ds_chunks	<code>section_content</code>

Table 3.4: MIMIC-IV log-tables: message content for each table. Fields are shown in `typewriter`, while static text (e.g., “findings”) is in normal font. For visualization purposes, some variable names have been shortened: `freq` = frequency, `sched` = `disp_sched`, `type` = `admission_type`, `loc` = `admission_location`, `ins` = insurance, `exp?` = `hospital_expire_flag`, `dtime` = `deathtime`.

individual sections and selecting those deemed relevant for the task of generating discharge instructions.

The process of dividing the discharge summary into its component sections is described in Section 3.4.3. As outlined there, this procedure produces the `ds_chunks` table, which is the sole source used to construct the patient summary.

The `ds_chunks` table contains one column for each section of the discharge summary. In the patient summary, the selected sections are included in the following order: Allergies, History of Present Illness, Social History, Family History, Brief Hospital Course, Discharge Medications, Discharge Disposition, Discharge Diagnosis, and Discharge Conditions. Each section is inserted with its name as a header, followed by its content. Sections that are empty in the original discharge summary are omitted.

In the context of MIMIC-IV, the patient summary thus consists essentially of a re-assembled discharge summary, with certain sections removed according to task relevance rationale.

3.4.7 Data Split: Train, Test, DB, Recollection

This section describes how MIMIC-IV admissions are split into **four subsets: train, test, db, and recollection test**. The split process is explicitly designed to prevent overfitting and data leakage, ensuring both fairness in evaluation and the privacy of BIDMC patients.

In developing machine learning models, two significant challenges that can lead to misleading performance are overfitting and data leakage. **Overfitting** happens when a model becomes too specialized to its training data, memorizing specific examples rather than grasping the underlying, generalizable patterns. This makes it perform poorly on new, unseen data. **Data leakage**, on the other hand, is a more subtle issue where information from outside the training dataset improperly influences the model’s learning. This can occur when there’s an unintentional overlap or correlation between the training and test sets, leading to an overly optimistic evaluation of the model’s true performance.

In the context of health records, these problems are amplified by **subject contamination**. If the same subject appears in both training and testing, it becomes unclear whether the model is generalizing to a new clinical case or simply recalling information from training. Even though MIMIC-IV is pseudonymized, pseudonymization alone does not eliminate the risk of subject contamination.

Consider the following scenario. A patient with type 2 diabetes is admitted for a leg fracture, during which their treatment includes insulin therapy and metformin. Later, the same patient is admitted for an ischemic stroke. In the clinical diary of the stroke admission, the ongoing use of insulin is not explicitly mentioned. However, because the model has already seen the patient during the fracture admission, it may recall that the patient is diabetic and infer the insulin therapy, even though this information is absent from the new record. The link can be made through stable features—such as the patient’s age, ethnicity, or past medical history—that are repeated across admissions. As a result, the model’s predictions may be influenced by recalled knowledge of the patient rather than by the actual content of the input record.

The model can learn about the patient if it sees them many times during training. Therefore, it is essential to **include at most one admission per subject in the training set**. For evaluation, it is preferable to rely on patients not seen during training, for the same reason. Secondary hospitalizations of training subjects can instead be used to check for this **recollection** effect. If performance on the recollection test set is inflated compared to the standard test, this may be a sign of overfitting due to data leakage.

The data split process of this work, depicted in Figure 3.7, is based on the *admissions* table from the MIMIC-IV-HOSP module. This table contains 331,793 admissions (`hadm_id`) corresponding to 145,793 unique patients (`subject_id`). Since each patient was admitted on average more than twice, without a careful strategy the same subject could easily appear in both the train and test sets. To address this problem, the data split process unfolds in two separate steps.

First, the `admissions` table is grouped by `subject_id`, and exactly one admission (`hadm_id`) is sampled from each group. This results in 145,914 admissions, one per subject, which will later be divided into train and test. All the remaining

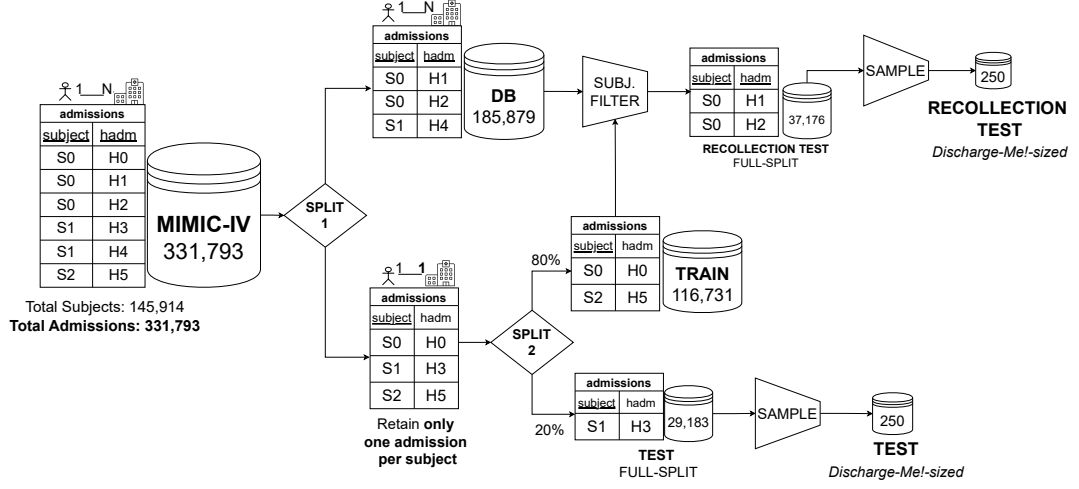


Figure 3.7: Representation of the data split process with a toy example. The example shows three subjects (S0, S1, S2) with different numbers of admissions. In the first step, one admission per subject is retained, while the remaining admissions are moved to the *db* set. The retained admissions are then divided into train (80%) and test (20%). Additional admissions from training subjects are later used to form the *recollection* set. Finally, the test set is downsampled to 250 examples to follow the *Discharge-Me!* framework.

admissions (185,879) are set aside in a separate partition called *db*.

Second, the set of **one-admission-per-subject** is split into train and test according to the standard 80/20 rule. The training set contains 116,731 admissions, while the test set contains 29,183. Following the *Discharge-me!* framework, only 250 randomly chosen test examples are used to rank GETALP models.

An additional evaluation set, called **recollection**, is built to check whether the model is overfitting on BIDMC patients. To construct it, the subjects included in the training set are identified, and their other admissions are retrieved from *db*. In this way, the recollection set contains new admissions belonging to patients the model has already seen during training. If performance on this set is skewed higher compared to the standard test set, it means the model is recalling patient-specific information instead of generalizing properly.

Each of the three splits—training, test, and recollection test—is filtered according to the availability of the required ground truth. Ground truth entries are considered valid as long as they are not empty; cases consisting solely of whitespace characters are excluded. For the dataset used to train on HC, an additional filtering step is applied using the log tables: entries with an empty **message** field are removed, and admissions with no remaining events are discarded, since they would otherwise yield

an empty diary. Likewise, the dataset for DI is filtered by excluding admissions without any clinical notes suitable for building a patient summary. For this reason, **train, test, and recollection splits are not of an equal size across tasks**, as reported in Table 3.5.

Split	Hospital Course	Discharge Instructions
Train	104,528	114,775
Test	26,102	28,664
Recollection	32,088	36,709

Table 3.5: Split size per task.

It is important to note that **no DS quality-based data cleaning** is performed, to ensure a fair comparison with the *Discharge-Me!* models.

3.4.8 MIMIC-IV Source Document Length

As anticipated in 3.1, for the design of source documents to be effective, the vast majority of documents should remain within the context window. The context window (also referred to as sequence length or model capacity) is the maximum number of tokens the model can process. Although models can technically process inputs beyond this limit without interruption, performance is not guaranteed beyond that threshold.

On average, a MIMIC-IV-Diary is 4,505 tokens long, which corresponds to about half of Mistral7B’s training context size (8k). The probability that a MIMIC-IV-Diary prompt exceeds the GETALP-Mistral7B sequence length (15k) is less than 2.9×10^{-2} , demonstrating that **the diary design effectively compresses the patient’s clinical history**.

By design, the patient summary is significantly shorter than the diary. Indeed, the MIMIC-IV-Patient Summary has an average sequence length of 1,531 tokens in the test split, with the maximum prompt length never exceeding 10k tokens in any split.

Finally, the length of the training prompt is analyzed; it consists of the concatenation of the input prompt and the ground truth. For HC, the training prompt is about 5,312 tokens on average, exceeding the GETALP-Mistral7B context window in 3.5% of cases (approximately 3,657 hospitalizations). **Models are not trained on examples exceeding their capacity**, since the ground truth would be truncated, potentially degrading performance. Discharge instructions training prompts never exceed the GETALP models’ context window.

Appendix B.2 reports tables and figures at the component level (source document, ground truth (HC, DI), training prompt).

3.5 Training GETALP Models

With the input design (Sections 3.1 and 3.2) and the datasets (Section 3.4) already defined, the next step consists in training domain-adapted language models. The **Asclepius family was selected as the clinical foundation model**, as it provides a high-performing and publicly available clinical model trained on synthetic discharge-style notes. Thanks to its open-source nature and clinically oriented pretraining, Asclepius enables domain-specific fine-tuning while avoiding the privacy limitations associated with real medical data. Starting from Asclepius-Mistral7B, **we derive GETALP-Mistral7B**, whereas Asclepius-Llama3-8B serves as the base for **GETALP-Llama3-8B**. Both models are fine-tuned using the QLoRA technique under equivalent configurations, with training performed in a supervised, completion-only fashion, as outlined in the following two subsections.

3.5.1 Supervised Fine-Tuning

As introduced above, SFT is performed in a **completion-only** fashion, meaning that model weights are updated solely based on the target outputs rather than the full content of the source documents. This preserves the model’s **flexibility** to adapt to different EHR structures while specializing in generating specific sections.

For each GETALP model, two task-specific sets of weights are trained:

- **HC**: trained on MIMIC-IV diaries (Section 3.4.5);
- **DI**: trained on MIMIC-IV patient summaries (Section 3.4.6).

Only the appropriate set of trained weights is loaded for inference, depending on the task.

The SFT loss for each task is formally expressed as:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^{T_i} \log P_{\theta}(y_t^{(i)} \mid y_{<t}^{(i)}, x^{(i)}),$$

where $x^{(i)}$ is the input prompt, $y_t^{(i)}$ is the t -th token of the target sequence, T_i is the number of tokens of $x^{(i)}$ length, and N is the number of training examples.

3.5.2 QLoRA Fine-Tuning

Quantized low rank adaptation is employed to fine-tune both Mistral7B and Llama3 efficiently. QLoRA consists of two main components:

- **Low-Rank Adaptation (LoRA)**: adds lightweight trainable adapters to projection and MLP layers, keeping the base model frozen. This drastically

reduces the number of parameters to train and avoids catastrophic forgetting. GETALP-Mistral7B adds only 41.9M trainable parameters ($\tilde{0.60\%}$ of the base model), while Llama3 adds 41.9M ($\tilde{0.52\%}$).

- **Quantization-aware training (4-bit):** the frozen weights are stored in 4-bit precision during training, minimizing VRAM usage and speeding up computation. Training is therefore feasible on smaller hardware without compromising the base model knowledge.

In summary, QLoRA provides several key advantages. It keeps the **total model size low**, preserves the original weights intact—thereby **preventing catastrophic forgetting and reducing the risk of overfitting**—and improves **training efficiency** in terms of both time and GPU memory usage. Training parameters, memory usage and learning curves are shown in Appendix C.2.

3.6 Evaluation Framework

This section details the evaluation framework used to benchmark GETALP models. The framework is taken from *Discharge Me: BioNLP ACL’24 Shared Task on Streamlining Discharge Documentation* [12].

It consists of eight metrics, namely **BLEU**, **ROUGE-1**, **ROUGE-2**, **ROUGE-L**, **METEOR**, **BERTScore**, **AlignScore**, and **MEDCON** [37, 38, 39, 40, 41, 42].

The evaluation is first conducted **task-wise**. Then, an **overall score** is computed by averaging the performance of the model across all metrics on both HC and DI sections.

As in *Discharge-Me!*, the model is evaluated on a set of **250 encounters** randomly sampled from the *test* split. As detailed in 3.4.7, an equal number of hospitalizations of subjects seen during the training phase is drawn from the *recollection* set, to test GETALP-Mistral7B against patient memorization.

The math behind metrics is available in the Appendix of this document (D).

3.6.1 Metrics

Metrics are defined in terms of a candidate and a reference text. In this report, the **candidate** is the target section generated by the AI assistant, while the **reference**—also called **ground truth**—is the corresponding section written by a human.

Although metrics can be generalized to multiple references per candidate, each hospitalization in our dataset includes only one version of target sections (one HC, one DI). Therefore, the description of the metrics is limited to the single-reference case. All metric scores range from 0 (worst) to 1 (best).

BLEU

BLEU (Bilingual Evaluation Understudy) is an algorithm originally introduced for evaluating the quality of machine translation [37]. BLEU is based on the concept of **n-grams**, which are **sequences of n consecutive words**. For example, the bi-grams of the sentence “He loves to read” are three:

He loves, loves to, to read.

The central idea behind BLEU is straightforward: count how many n-grams of the candidate sentence belong to the reference. However, this idea requires some refinements for stabilization:

- **clipping**: if an n-gram appears at most m times in the reference, it can be counted no more than m times. This step produces the so-called modified n-gram precision.
- **brevity penalty**: if the candidate is shorter than the reference, a brevity penalty BP is applied.
- **geometric mean**: BLEU-4 is obtained by taking the geometric mean of the modified precisions of uni-grams, bi-grams, tri-grams, and tetra-grams.

BLEU-4 shows the highest alignment with human judgment [37].

ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a family of metrics designed to evaluate the **overlap** between a candidate text and a reference text [38].

The variants of ROUGE used in this report are:

- **ROUGE-1** (R-1): based on uni-grams. It measures how many single words from the reference are correctly reflected by the candidate.
- **ROUGE-2** (R-2): based on bi-grams. It evaluates how many consecutive word pairs from the reference appear in the candidate.
- **ROUGE-L** (R-L): based on the longest common subsequence (LCS), which captures the longest sequence of words appearing in both texts while preserving order, even if the words are not strictly consecutive.

METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering) is a metric originally proposed for machine translation evaluation, as an improvement of BLEU [39]. Unlike BLEU, which is precision-oriented, METEOR was designed to balance both precision and recall, while also accounting for linguistic phenomena that go beyond exact word matching.

The key innovations of METEOR are:

- **Flexible matching:** words are matched not only by exact form, but also through **stemming** (“running” → “run”), **synonyms** (“doctor” ↔ “physician”), and **paraphrases** when available.
- **Precision-recall balance:** the final score is based on a harmonic mean that gives more weight to recall, reflecting the intuition that **missing important information is worse than adding redundant words**.
- **Fragmentation penalty:** if matches are scattered across the candidate and reference in a disordered way, the score is penalized. This captures **fluency** and word ordering better than n -gram overlap alone.

These properties make METEOR particularly suitable in our clinical context, where it is preferable for discharge documentation to be somewhat redundant rather than omitting critical medical information, and where many acronyms and synonyms may be encountered.

BERTScore

BERTScore is a **semantic similarity** measure that leverages contextual embeddings from pretrained language models to compare candidate and reference texts [40]. Unlike BLEU or ROUGE, it captures meaning rather than exact word overlap, making it particularly suitable for clinical text, where synonymous expressions may convey identical medical information.

In this project, we used the `distilbert-base-uncased` model, computing embeddings from the fourth layer out of six available. As discussed by Peters et al. and Tenney et al., the inner layers of BERT tend to focus more on syntactic information, while the higher layers capture semantic information. Although the motivations behind the specific choice of the fourth layer by the authors of the *Discharge-Me!* challenge are not publicly disclosed, this choice is convincing: it positions BERTScore as a **middle ground** between metrics more focused on lexical overlap, such as BLEU and ROUGE, and metrics more semantically oriented, such as AlignScore and MEDCON, the latter being specifically designed for medical text.

The measure calculates token-level cosine similarities between candidate and reference embeddings that are then aggregated into precision, recall, and finally F1-score, as shown in Figure 3.8.

In Chapter 4, tables report “BERT” as an abbreviation of BERTScore.

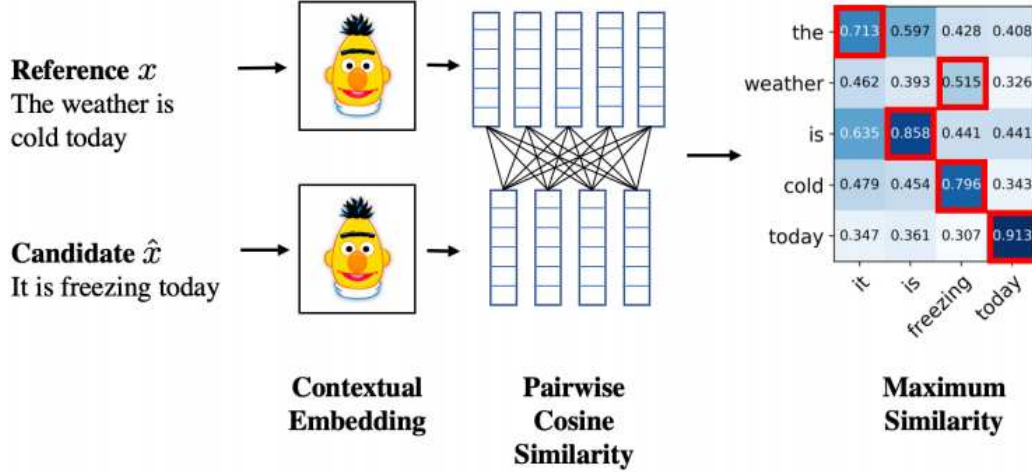


Figure 3.8: Example of BERTScore computation from Zhang et al.[40]. Each sentence is first encoded through a BERT model (distilbert-base-uncased, 4th layer in this case) then the cosine similarity matrix is computed. In red, values used to compute the recall. The BERTScore is calculated as the F1-Score: the harmonic mean of precision and recall.

AlignScore

AlignScore is a semantic evaluation score designed to measure the **factual consistency** of generated text with respect to a reference [41]. It is particularly suitable for clinical text, where preserving factual information is critical.

The candidate text is split into sentences and the reference text into chunks of roughly 350 tokens. Each candidate sentence is compared with all reference chunks using a pretrained NLI model. Here, the alignment function corresponds to the entailment probability between the candidate sentence and a reference chunk.

For each candidate sentence, the maximum entailment probability across all reference chunks is selected. The AlignScore is then computed as the mean of these maximum probabilities across all candidate sentences (see Figure 3.9).

AlignScore **cannot be considered a metric** in the strict mathematical sense, as it fails to satisfy two essential properties: symmetry and identity. For the sake of simplicity, it will still be called a metric.

In Chapter 4, tables report “Align” as an abbreviation of AlignScore.

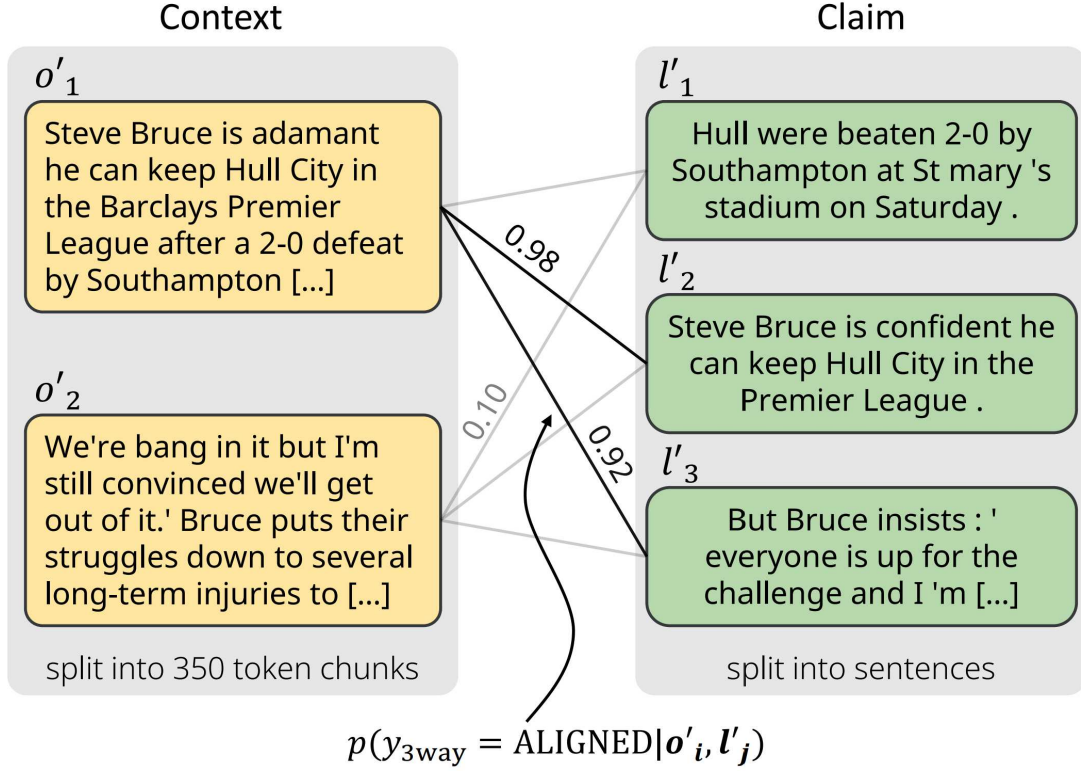


Figure 3.9: Example of AlignScore computation from Zha et al.[41]. The context (reference) is split into roughly 350-token chunks. Then, each sentence in the claim (candidate) is evaluated against the context chunks using the alignment function. The highest alignment score of each claim sentence is selected and then averaged to derive the factual consistency score.

MEDCON

MEDCON is a medical concept-based evaluation metric designed to assess the **accuracy and consistency of clinical concepts** by comparing the presence of key medical concepts in the candidate text against those in the reference [42].

To ensure clinical relevance, the evaluation is restricted to specific UMLS semantic groups, including Anatomy, Chemicals & Drugs, Devices, Disorders, Genes & Molecular Sequences, Phenomena, and Physiology.

The metric then computes the F1-score between the sets of extracted concepts, providing a measure of how well the candidate captures the reference’s clinical content.

3.6.2 Per-Task Evaluation

For each metric β , the score is computed separately on the HC and DI sections. Let $S_{\beta,HC}$ be the score of metric β on the HC section, and $S_{\beta,DI}$ the score on the DI section.

The task-wise evaluation produces two aggregated values:

$$AVG_{HC} = \frac{1}{N} \sum_{i=1}^N S_{\beta_i,HC} \quad AVG_{DI} = \frac{1}{N} \sum_{i=1}^N S_{\beta_i,DI} \quad (3.1)$$

where $N = 8$ is the number of metrics (i.e., BLEU, ROUGE-1,-2,-L, METEOR, BERTScore, AlignScore, MEDCON).

Thus, AVG_{HC} represents the average performance of the model on the HC section across all metrics, while AVG_{DI} represents the corresponding average for the DI section. These two averages allows to identify **strengths and weaknesses of the model in each task** before computing the overall score.

In Chapter 4, tables report this score in the column “AVG”.

3.6.3 Overall Scoring System

To obtain a comprehensive score, results are aggregated across tasks and metrics. For each metric β , the mean of HC and DI scores is first computed:

$$\beta = \frac{S_{\beta,HC} + S_{\beta,DI}}{2} \quad (3.2)$$

Then, the overall score is calculated as the mean of all metric scores:

$$\text{Overall} = \frac{1}{N} \sum_{i=1}^N \beta_i \quad (3.3)$$

The overall score is used to elect the best stystem, in this study as in *Discharge-Me!*.

3.7 Experiments

This section presents the set of five experiments conducted in this study. Each experiment is described in detail in the following subsections, with the aim of ensuring **methodological clarity and interpretative consistency**. The experiments cover different aspects of model evaluation, from internal comparisons of fine-tuned models to ablation studies and qualitative assessments of the reference data.

All experiments are evaluated using the framework described in Section 3.6, ensuring systematic and reproducible results. A single *test* dataset of 250 hospitalizations, created according to the data splitting pipeline described in 3.4.7, is

consistently used across all experiments. Both HC and DI generation tasks are evaluated on the identical set of hospitalizations. In the recollection test, this standard *test* set is still employed, while an additional *recollection* set is used to investigate o specifically determine whether the model memorizes patient-specific information.

3.7.1 Evaluation of GETALP Models

The experiments are designed to evaluate the models developed in this work, GETALP, in a systematic and comparable manner. Three main analyses are performed within this experimental setup:

1. **GETALP vs. Asclepius baselines.** Each fine-tuned GETALP model is compared against its corresponding base model, Asclepius, to assess whether the fine-tuning procedure introduces measurable improvements over the clinical foundation models.
2. **Comparison between GETALP-Mistral7B and GETALP-Llama3-8B.** A head-to-head evaluation of the two fine-tuned models developed in this study is conducted to identify the best-performing solution among the models produced in this work.
3. **Benchmarking against external solutions.** The two fine-tuned GETALP models are compared with the systems presented in the *Discharge-Me!* challenge, allowing a direct assessment of performance against other state-of-the-art clinical AI-assistants.

This experimental design enables a systematic evaluation of the models, addressing both improvements due to fine-tuning and the relative performance against external competitive solutions.

Results are available in Section 4.1.

3.7.2 Ablation Study on the Diary Content

The ablation study systematically evaluates the contribution of different **input components** to a model’s predictions by selectively removing or modifying them. In this work, the ablation study focuses on the content of the MIMIC-IV-Diary, assessing how the removal of specific sources of information affects the quality of the hospital course generated by the model.

The diary content is described in Section 3.4.5, and, as previously introduced, it is structured as a sequence of log entries, each consisting of a triplet [timestamp-action] message. Ablation is performed by selectively **removing**

specific **actions**, creating different versions that allow systematic assessment of how the presence or absence of particular types of information impacts the quality and completeness of the hospital course generated by the model.

GETALP-Mistral7B is used as the reference model, as it outperformed Llama3-8B in terms of overall score (see Section 4.1.3). Importantly, all ablation experiments are performed using GETALP-Mistral7B trained on the complete MIMIC-IV-Diary: **no variant-wise training is performed**.

Information in the diary can be split into two categories according to their source:

- **structured data**: automatically collected records or entries compiled with minimal effort by clinicians, precisely those of MIMIC-IV-HOSP and MIMIC-IV-ED described in Table 3.1 and in Section 3.4.1. These records are concise, standardized, and represent fundamental elements of the hospital stay.
- **clinical notes**: narrative documents written directly by clinicians, who must sit down, reflect on the patient’s case, organize the information, and express it in a coherent style. Sourced from MIMIC-IV-NOTE, these consist of radiology reports (see Section 3.4.3) and the narrative sections extracted from DSes, detailed in Table 3.2.

The key distinction lies in the **degree of human effort** required: structured data are generated with minimal manual input, whereas clinical notes require clinicians to interpret, organize, and communicate the information.

Based on these considerations, the following configurations are defined:

- **Complete**: baseline, all information available. The diary is made of logs as described in Tables 3.3 and 3.4 within Section 3.4.5.
- **No Radiology (-R)**: the classic MIMIC-IV-Diary excluding radiology notes. As noted by Liu et al., radiology reports occasionally introduce spurious details that may lead to hallucinations, and their removal is expected to slightly improve performance [101].
- **Clinical Notes Only (-HOSP,-ED)**: retains only narrative clinical notes, excluding radiology. In practice, this corresponds to retaining only the DSes. In terms of content, this configuration is similar to the input proposed by Damm et al., *Discharge-Me!* winners. It is important to note that a diary composed solely of clinical notes would render any effort to automate HC generation largely ineffective. Again, to effectively relieve clinicians from the discharge documentation burden, the system must rely as little as possible on human-written content.

- **Clinician-Free Baseline (-DS / -DS,+DS)**: configurations designed to minimize direct clinician involvement in generating the hospital course. These settings include only structured data and potentially automatable notes. The **-DS** configuration removes all content extracted from discharge summaries using the DSChunker algorithm (see Section 3.4.3), while **-DS,+DS** reintroduces solely the history of present illness. This choice reflects the high informational value of the HPI, which conveys the core clinical context of the hospitalization, and aligns with future perspectives of ASR automation pipelines described in the final chapter of this work (6). Radiology notes are also consistently preserved, in line with ongoing research on automatic report generation (RRG24 [11]).

Each configuration is labeled according to the components removed: the “-” symbol precedes the name of any section that has been excluded, while the “+” symbol indicates a section that has been selectively reintroduced.

Results are available in Section 4.2.

3.7.3 Using AI-Generated Hospital Courses for Discharge Instruction Generation

In this experiment, **discharge instructions are generated using the hospital course produced by GETALP-Mistral7B itself ($\widehat{HC}$)**. In other words, unlike other experiments where the model relies on the human-written HC, here GETALP-Mistral7B generates DI based on its own prior output.

The motivation stems from how a clinical AI assistant would be deployed in real hospital wards. To produce discharge instructions, the model relies on the patient summary, which includes, among other sections, the patient’s hospital course. Since GETALP-Mistral7B can also generate HC, a realistic workflow would involve first generating $\widehat{HC}$ from the patient’s diary and then using it to generate DI.

From a technical standpoint, patient summaries in the test set are modified by **replacing** the clinician-written HC with the model-generated $\widehat{HC}$. The model then generates DI from these modified summaries. We denote this configuration as $DI(\widehat{HC})$, in contrast to the baseline $DI(HC)$, which uses the original physician-authored HC.

This cascading setup serves two purposes. First, it evaluates the **feasibility** of a fully automated generation pipeline under realistic conditions. Second, it provides an indirect **quality assessment** of GETALP-Mistral7B’s hospital courses, by examining how its content affects downstream DI performance.

Results for this experiment are presented in Section 4.3.

3.7.4 Qualitative Analysis of Model Outputs

A complementary qualitative analysis is conducted to systematically examine the outputs generated by the models. This assessment is performed **manually**, reviewing the texts produced by the GETALP models.

The purpose of this analysis is to identify potential issues in the generative process, such as inconsistencies, omissions, or clinically implausible statements, and to explore possible directions for improvement. By examining individual outputs in detail, it is possible to highlight recurring patterns of errors, evaluate the strengths and weaknesses of the model, and gain insights that cannot be captured by automatic metrics alone.

Such an approach complements quantitative evaluation and provides a more comprehensive understanding of the model’s behavior in practical clinical scenarios.

Results are available in Section 4.4.

3.7.5 Recollection Test

The recollection test is designed to evaluate whether GETALP-Mistral7B **memorizes** patient-specific information from the training data.

For this experiment, the model is evaluated on a subset of 250 hospitalizations corresponding to patients from the BIDMC who were **included in the training set**.

If the model demonstrates substantially higher performance on this subset compared to a test set consisting of previously **unseen** patients, this would indicate memorization of patient-level information. Such a result would imply the presence of overfitting, with direct implications for privacy and data leakage, entailing the need for adjustments in the training pipeline. The detailed procedure used to construct the recollection test set is described in Section 3.4.7.

Results are available in Section 4.5.

Chapter 4

Results

Models are evaluated using the *Discharge-Me!* framework introduced in Section 3.6. Results from the *Discharge-Me!* competition are taken from the official leaderboard, accessible at stanford-aimi.github.io/discharge-me. At the time of writing, the leaderboard was last updated on May 12, 2024. These results are also reported and discussed in Xu and PhysioNet [12, 11].

The organization of this chapter follows the same logical flow as the experimental design in Section 3.7. Each major section here corresponds to one of the experimental subsections from the Methods chapter, maintaining the same sequence. To facilitate cross-referencing, each results section begins by recalling the corresponding experimental setup, creating a direct mapping between methodological choices and their outcomes.

4.1 Evaluation of GETALP Models

The description of the experiment is available at Section 3.7.1.

4.1.1 GETALP vs Asclepius

The fine-tuning process leads to **substantial improvements for both architectures**, as evidenced by the overall scores presented in Table 4.1, which compares the Mistral7B and Llama3-8B models before (Asclepius) and after fine-tuning (GETALP).

For Mistral7B, the baseline model, i.e. Asclepius-Mistral7B, achieves an overall score of 0.138, while the fine-tuned version, GETALP-Mistral7B, reaches 0.393, resulting in a gain of 0.255. Similarly, for Llama3-8B, Asclepius-Llama3-8B starts from 0.148, and GETALP-Llama3-8B rises to 0.370, with an improvement of 0.222.

The improvements are also observed at the level of individual tasks, as shown in

Rank	Team	Overall	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
1	GETALP-M7B	0.393	0.203	<u>0.482</u>	0.287	0.367	0.487	0.402	0.447	0.471
2	GETALP-L8B	<u>0.370</u>	<u>0.169</u>	0.490	<u>0.254</u>	<u>0.333</u>	<u>0.464</u>	<u>0.393</u>	<u>0.415</u>	<u>0.444</u>
3	Asclepius-L8B	0.148	0.005	0.236	0.047	0.130	0.205	0.135	0.238	0.189
4	Asclepius-M7B	0.138	0.004	0.214	0.037	0.121	0.175	0.117	0.270	0.164

Table 4.1: Overall Scoring System - GETALP vs Asclepius. Best scores in **bold**, second-best scores underlined. Acronyms: M7B = Mistral7B, L8B = Llama3-8B.

Table 4.2. Considering the average metric score AVG, Mistral7B increases by 0.165 on HC and 0.346 on DI, while Llama3-8B improves by 0.151 on HC and 0.293 on DI.

Model	AVG	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
Hospital Course (HC)									
Mistral7B									
Asclepius-M7B	0.144	0.003	0.214	0.045	0.124	0.188	0.106	0.289	0.183
GETALP-M7B	<u>0.309</u>	0.127	<u>0.411</u>	0.198	0.281	0.424	<u>0.307</u>	0.351	<u>0.375</u>
Llama3-8B									
Asclepius-L8B	0.161	0.006	0.250	0.060	0.142	0.230	0.126	0.260	0.218
GETALP-L8B	0.312	<u>0.116</u>	0.462	<u>0.195</u>	<u>0.274</u>	0.424	0.329	<u>0.322</u>	0.378
Discharge Instructions (DI)									
Mistral7B									
Asclepius-M7B	0.131	0.004	0.213	0.030	0.118	0.163	0.128	0.250	0.144
GETALP-M7B	0.477	0.279	0.552	0.376	0.453	0.550	0.496	0.543	0.568
Llama3-8B									
Asclepius-L8B	0.135	0.004	0.222	0.034	0.119	0.179	0.144	0.216	0.159
GETALP-L8B	<u>0.428</u>	<u>0.222</u>	<u>0.518</u>	<u>0.313</u>	<u>0.391</u>	<u>0.505</u>	<u>0.457</u>	<u>0.508</u>	<u>0.509</u>

Table 4.2: Per-Task Results. Best scores in **bold**, second-best scores underlined. Acronyms: M7B = Mistral7B, L8B = Llama3-8B.

Interestingly, the ranking of the two architectures **inverts** after fine-tuning: prior to fine-tuning, Llama3-8B outperforms Mistral7B on nearly all metrics regardless of the task considered, whereas post fine-tuning **Mistral7B** surpasses it, demonstrating a **stronger adaptation** to the generation tasks in object (HC and DI).

4.1.2 GETALP-Mistral7B vs GETALP-Llama3-8B

The head-to-head comparison between the two fine-tuned models highlights complementary strengths, as reported in Tab. 4.2. For the **HC** section, **GETALP-Llama3-8B wins** by an AVG marginal 0.003 points, with the most notable advantage in ROUGE-1 (+0.049). However, GETALP-Mistral7B achieves higher

scores in four of the eight metrics, including BLEU, ROUGE-2, ROUGE-L, and AlignScore. Both models tie on BERTScore, while Llama3-8B has a slight edge in MEDCON (+0.003), suggesting it is slightly better at capturing the clinical concepts reported by doctors in their hospital courses.

For **DI**, **GETALP-Mistral7B** consistently outperforms GETALP-Llama3-8B across all metrics, with an AVG margin of 0.049—more than sixteen times the advantage that Llama3-8B holds over Mistral7B in the HC section.

Looking at the overall score (Tab. 4.1), the balance tilts in favor of GETALP-Mistral7B, which reaches an overall score of 0.393 compared to GETALP-Llama3-8B’s 0.370. This indicates that, despite Llama3-8B maintaining a slight advantage for HC summaries, **GETALP-Mistral7B emerges as the most effective solution developed in this work.**

4.1.3 GETALP vs *Discharge-Me!*

Rank	Team	Overall	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
1	GETALP-M7B	0.393	0.203	<u>0.482</u>	0.287	0.367	0.487	0.402	0.447	0.471
2	GETALP-L8B	<u>0.370</u>	<u>0.169</u>	0.490	<u>0.254</u>	<u>0.333</u>	<u>0.464</u>	0.393	<u>0.415</u>	<u>0.444</u>
3	WisPerMed	0.332	0.124	0.453	0.201	0.308	0.438	<u>0.403</u>	0.315	0.411
4	HarmonAI	0.300	0.106	0.423	0.180	0.284	0.412	0.381	0.265	0.353
5	aehrc	0.297	0.097	0.414	0.192	0.284	0.383	0.398	0.274	0.332
6	EPFL-MAKE	0.289	0.098	0.444	0.155	0.262	0.399	0.336	0.255	0.360
7	UF-HOBI	0.286	0.102	0.401	0.174	0.275	0.395	0.289	0.296	0.355

Table 4.3: Updated *Discharge-Me!* leaderboard, including GETALP models. Best scores in **bold**, second-best scores underlined. Acronyms: M7B = Mistral7B, L8B = Llama3-8B.

Table 4.3 shows the overall leaderboard of the *Discharge-Me!* challenge, including both GETALP-Mistral7B and GETALP-Llama3-8B. **GETALP models outperform all other participants** across nearly all evaluated metrics, achieving overall scores of 0.393 and 0.370, respectively.

WisPerMed, ranked third after inserting GETALP solutions in the leaderboard, was the official winner of the *Discharge-Me!* challenge, surpassing all other participants on all metrics. Despite WisPerMed employing the same base models (Asclepius), the same fine-tuning approach (QLoRA), and equivalent training parameters as the GETALP solutions, both GETALP models achieve substantially higher scores across all metrics except METEOR, where GETALP-Llama3-8B ranks slightly below WisPerMed. In particular, **GETALP-Mistral7B**, the best-performing model, **surpasses WisPerMed across all metrics of the evaluation framework**, with a margin 0.061 in overall score.

Given that WisPerMed and GETALP share the same underlying model and fine-tuning setup, this performance gap is unlikely to result from the base model or

training procedure alone. It can be hypothesized that the **key factor lies in the careful design of diary and patient summary** layouts, each explicitly crafted to favor the characteristics of HC and DI generation, likely enabled the models to produce higher-quality outputs. In other words, the difference in performance highlights the importance of task-specific prompt engineering and source document design in automating discharge documentation.

Due to the non-disclosure of task-level results from the challenge, a per-task comparison between GETALP models and other *Discharge-Me!* participants cannot be provided. Regardless, the overall results clearly demonstrate that **GETALP-Mistral7B represents the state-of-the-art model for generating discharge documentation**.

4.2 Ablation Study on the Diary

The experimental setup is described in Section 3.7.2

Results of the ablation study are presented in Table 4.4.

Config	AVG	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
Complete	<u>0.309</u>	0.127	0.411	0.198	0.281	0.424	0.307	0.351	0.375
-R	0.310	0.128	0.410	0.202	0.284	<u>0.420</u>	<u>0.303</u>	0.358	<u>0.373</u>
-HOSP,-ED	0.308	0.123	<u>0.416</u>	<u>0.203</u>	<u>0.283</u>	<u>0.420</u>	0.298	<u>0.352</u>	0.370
-DS,+HPI	0.291	0.118	0.390	0.188	0.269	0.402	0.284	0.339	0.335
-DS	0.251	0.084	0.355	0.153	0.240	0.360	0.241	0.313	0.265

Table 4.4: Ablation study results on the hospital course generation task. Best scores in **bold**, second-best underlined. Model: GETALP-Mistral7B.

Findings seems to confirm those of Liu et al. (*aehtc*): radiology reports can act as noise. In facts, excluding them from the diary (-R) yields a slight improvement (+0.001 AVG). However, the complete configuration is more medical consistent (-0.002 MEDCON) and more semantically aligned with the ground truth (-0.004 BERTScore), indicating that **radiology notes still provide valuable information** to GETALP-Mistral7B. In light of such results, and considering that radiology reports are among the longest documents in the model input (Appendix B.2), a promising approach is not to remove them, but rather to summarize them into their essential components, such as examination details and findings.

From Table 4.4 it can be observed that retaining only clinical notes (-HOSP,-ED) outperforms relying solely on structured data (-DS) by 0.057, indicating the substantial informational **value of clinician-authored documentation**. When the HPI is re-introduced alongside structured data (-DS,+HPI), this gap is largely reduced to 0.017, corresponding to a gain of 0.04 over -DS. These findings highlight

the **HPI as a critical component**, emphasizing the potential value of developing an AI-assisted HPI generation pipeline.

In summary, the **best-performing diary configuration** is the complete version without radiology reports (**-R**), achieving an average score of 0.310. The **-DS,+HPI** setting, designed to remove clinician intervention, achieves 0.291, remaining only 0.011 points below the top result.

4.3 Using AI-Generated Hospital Courses for Discharge Instruction Generation

A detailed description of the experimental setup can be found in Section 3.7.3.

Overall, the performance of DI **decreases when using AI-generated HCs**. Specifically, the AVG score drops from 0.477 for $DI(HC)$ to 0.419 for $DI(\widehat{HC})$, and this difference is statistically significant ($p = 0.013$). This indicates that the observed decline is unlikely to be due to chance.

To investigate whether this decline is caused by variability in the quality of generated HCs, the $DI(\widehat{HC})$ samples were divided into two groups based on the median DI score, resulting in a *low-DI* and a *high-DI* group, each containing 125 hospitalizations. The AVG scores of the corresponding $\widehat{HC}$ were then compared between these two groups. The $\widehat{HC}$ AVG scores are 0.365 for the *high-DI* group and 0.250 for the *low-DI* group, with a statistically significant difference ($p < 10^{-7}$). This provides strong evidence that **the decrease in DI performance is primarily driven by a lower quality of the generated hospital courses**.

These findings are further illustrated in Figure 4.1. When the original clinician-authored HC is used as input, the distribution of average DI scores exhibits three prominent peaks around 0.3, 0.8, and 1. When AI-generated HC is used, the peak around 0.3 becomes higher and sharper, encompassing more than 25 examples, while the other peaks around 0.8 and 1 are no longer observed. This pattern confirms that **the quality of the HC directly influences DI validity**.

While the results suggest that $\widehat{HC}$ are not fully sufficient to support optimal DI production, it remains informative to evaluate the pipeline using the overall scoring system. The results are reported in Table 4.5.

When comparing leaderboard performance, the end-to-end GETALP-Mistral7B- $\widehat{HC}$ variant—where DI are generated from model-produced $\widehat{HC}$ —achieves an overall score of 0.364. This is slightly lower than the 0.393 obtained when using the original clinician-authored HC, but still surpasses the score of *WisPerMed* (0.332), the official winner of the *Discharge-Me!* Challenge [96, 11].

This result demonstrates that **the system maintains its advantage**—even under a fully automated pipeline relying exclusively on generated HC and DI, it outperforms all non-GETALP solutions.

Rank	Team	Overall	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
1	GETALP-M7B	0.393	0.203	0.482	0.287	0.367	0.487	0.402	0.447	0.471
2	GETALP-L8B	0.370	0.169	0.490	0.254	0.333	0.464	0.393	0.415	0.444
3	GETALP-M7B- $\widehat{HC}$	0.364	0.176	0.450	0.257	0.337	0.461	0.374	0.416	0.443
4	WisPerMed	0.332	0.124	0.453	0.201	0.308	0.438	0.403	0.315	0.411

Table 4.5: *Discharge-Me!* leaderboard including end-to-end pipeline. Acronyms and symbols: M7B = Mistral7B, L8B = Llama3-8B, $\widehat{HC}$ = AI-generated Hospital Course.

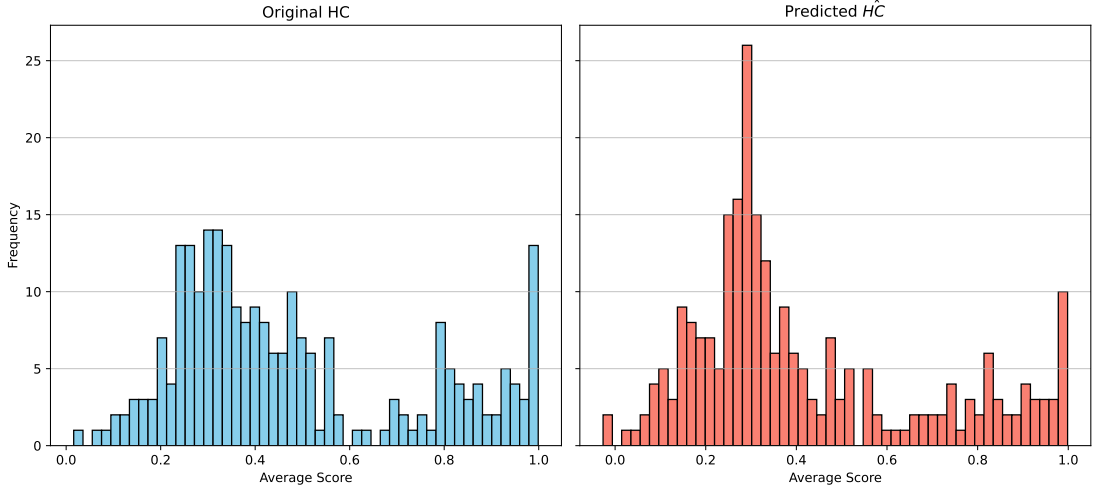


Figure 4.1: Distribution of average scores (AVG) on the DI task, when the input is the original HC (blue) or the AI-generated one (orange). The distribution changes significantly, with a peak around 0.3 that becomes higher and sharper, while the peaks around 0.8 and 1 disappear.

4.4 Qualitative Analysis of Model Outputs

This section presents a qualitative analysis of the outputs generated by GETALP-Mistral7B. The experimental setup is detailed in Section 3.7.4.

The model can produce **coherent and well-organized** hospital courses. Information is structured by clinical systems (e.g., cardiovascular, pulmonary, gastrointestinal), creating **readable** summaries. Key vitals and laboratory results are usually correctly reported, ensuring the output aligns with the source data. Post-operative information, when present, is correctly reported in the relevant sections. Furthermore, the **chronological sequence** of events is accurately **preserved**, demonstrating that **the diary effectively guides the model** in capturing the

temporal progression of the patient’s hospitalization. Sentences are **complete** and verb tenses are generally correct, contributing to **clarity** and coherence throughout the document.

In some cases, the model **infers** clinically relevant details from context (e.g., adding spirometry instructions), even though these are not explicitly stated in the diary. When these inferences are **consistent** with the ground truth, it indicates that the model is able to capture the underlying clinical trajectory. By reconstructing methodological details from the sequence of events—without requiring them to be explicitly stated—the model demonstrates a good **understanding** of the rationale behind the clinical management.

When it comes to discharge instructions, GETALP-Mistral7B produces texts that are **correctly addressed** to patients or, in the case of deceased patients, to their families. As expected, the language is less technical and more comprehensible, effectively translating clinical information into patient-friendly terms. Medications and recommended care measures are reported in accordance with the ground truth, despite occasionally misdosages.

Notably, GETALP-Mistral7B occasionally produces outputs that are **more detailed than the original** clinical notes. It is worth noticing that in such cases, the inference may be evaluated as a poor quality one, because it does not match the ground truth, even if it is better in terms of completeness and correctness.

Overall, **DI outputs appear to be of higher quality than HC**, as captured by the evaluation framework.

The main weaknesses of GETALP-Mistral7B are **hallucinations**, **degeneration** and **redundancy**, which are particularly prevalent in the HC task.

4.4.1 Hallucinations

The model occasionally introduces information not present in the diary, such as unprescribed medications, incorrect dosages, or fictional readmissions and procedures. These hallucinations are more frequent in HC, likely due to the presence of **gaps** in MIMIC-IV, which lacks certain clinical information typically available to clinicians while writing discharge summaries.

4.4.2 Degeneration

Degeneration occurs when the model repeatedly generates the same phrase or sequence until reaching the maximum token limit (Figure 4.2)

This phenomenon can involve not only single sentences but also entire blocks of text, alternating in an **endless loop**. Degeneration often starts after correct and complete information, producing sequences that are coherent at first but eventually get stuck in repetitive loops rather than being purely nonsensical. Numbers in

90% stenosis of LAD, 90% stenosis of LAD, 90% stenosis of LAD, 90% stenosis of LAD, 90% stenosis of LAD, 90% stenosis

Figure 4.2: Example of degeneration in HC output. The model gets stuck repeating the same phrase until reaching the token limit, that in this example is reached after *stenosis*.

these loops sometimes increment across repetitions (e.g., #POD:1, #POD:2), and the behavior is particularly frequent while writing the Transitional Issues section of an hospital course.

Manual inspection of all generated examples shows that degeneration is especially prevalent in HC, occurring in approximately 29.27% of cases, predominantly among lower-ranked outputs. By contrast, DI outputs are rarely affected, likely because they end with a predictable “Follow-up” section, which helps the model terminate correctly.

It is hypothesized that this **degeneration largely influences** the difference in terms of AVG score between tasks, with DI outperforming HC by 0.168 points (0.477 vs 0.309, see Table 4.2). A cumulative distribution of degenerative examples is provided in Figure 4.3. Hospital courses were first ordered by AVG score from lowest (rank 1) to highest (rank 250). Degeneration was manually labeled: an example is considered degenerated if it repeats the same sentence continuously until reaching the generation token limit, 1000. Results show that half of degenerated examples fall within the first 56 ranks (66%), while more than the 78% of the 23 lowest-ranked HC outputs are degenerated—18 examples. Therefore, while degeneration is mostly concentrated among low-quality HCs, a non-negligible number of degenerated examples appear even at higher ranks, indicating problematic aspects in the generation process.

Degeneration Absence in Llama3-8B

Notably, **degeneration was not observed in GETALP-Llama3-8B**. Despite undergoing the same fine-tuning process, this model consistently produced outputs that terminated appropriately. It is therefore unsurprising that the *Discharge-Me!* automatic evaluation framework assigned higher scores to the HCs generated by Llama3. Inference parameters were kept equivalent across models (`temperature = 1`, `max_new_tokens = 1000`), ruling out decoding settings as a cause.

These findings suggest that **degeneration is more likely linked to architectural aspects** of the Mistral7B model rather than the training protocol or generative setup. Such an observation points either to a stronger competitiveness

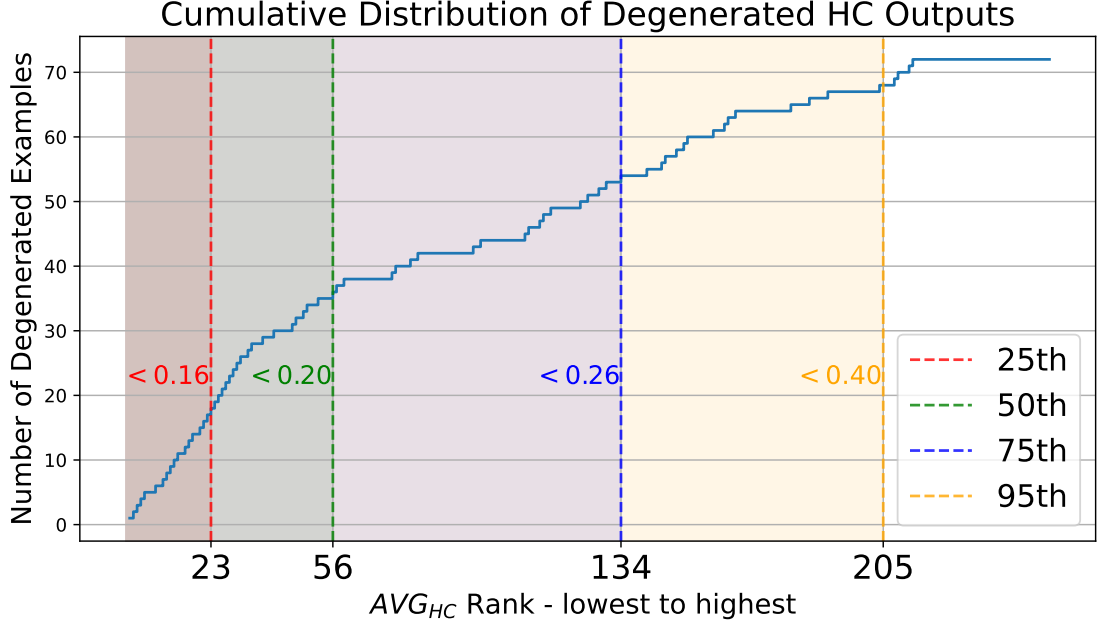


Figure 4.3: Cumulative distribution of degenerated examples in the hospital course task. The x -axis reports the rank positions of the HCs, ordered from lowest to highest according to the *Discharge-Me!* score, while the y -axis indicates the cumulative number of degenerated outputs observed up to that rank. Vertical dashed lines mark the 25th, 50th, 75th, and 95th percentiles of the distribution, and the shaded areas highlight the portions of the ranking that fall below each percentile. The labels report the score thresholds ($< AVG_{HC}$) corresponding to these cutoffs. The figure shows that degeneration is mainly concentrated among low-quality HCs, yet a non-negligible number of degenerated examples appear even at higher ranks, indicating that degeneration is not exclusively associated with poor scoring outputs.

of Llama3 for text generation tasks, or to advantages conferred by its pre-training strategy at the foundation model stage.

Mitigation Strategies

The approach used to reduce degeneration is the application of **n-gram penalties**, which discourage the model from repeating sequences of n tokens. Figure 4.4 shows that setting `n_grams_penalty = 2` reduces the frequency of very low-scoring HC. However, applying such penalties can have unintended consequences by altering the model’s output behavior.

In particular, introducing an n -gram penalty increases the model’s **creativity**,

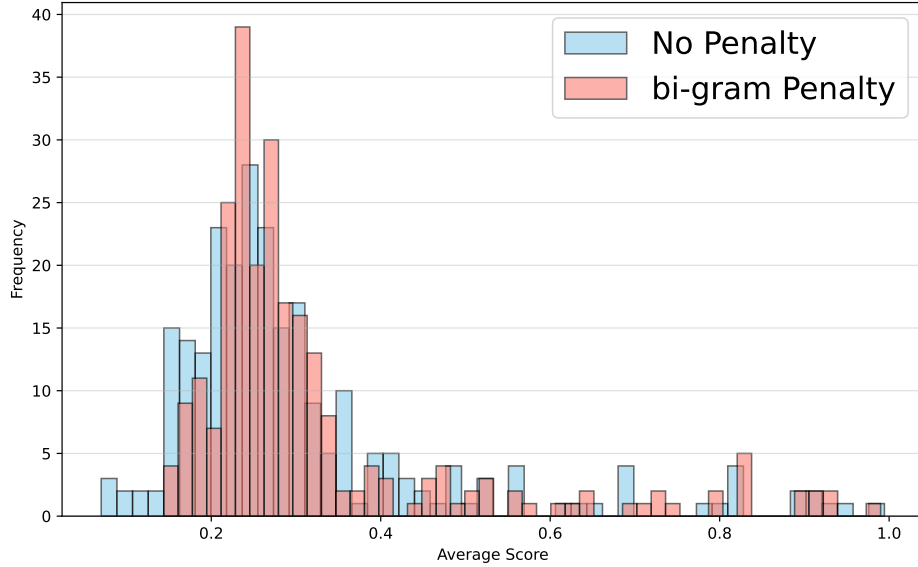


Figure 4.4: N-gram penalty: histogram of average score (AVG) on the HC task. Bi-gram penalty reduces the frequency of very low scores < 0.1 . It is worth noticing that the *Discharge-Me!* framework has no metric that explicitly punishes for hallucinations.

n-gram	Overall	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
0	0.309	0.127	0.411	0.198	0.281	0.424	0.307	0.351	0.375
2	0.325	0.127	0.457	0.213	0.301	0.439	0.318	0.354	0.387
3	0.265	0.069	0.411	0.151	0.239	0.387	0.261	0.307	0.295
4	0.297	0.099	0.437	0.189	0.278	0.420	0.284	0.343	0.329
6	<u>0.320</u>	<u>0.122</u>	<u>0.453</u>	<u>0.208</u>	<u>0.297</u>	<u>0.435</u>	<u>0.313</u>	<u>0.352</u>	<u>0.380</u>

Table 4.6: Hospital Course – n-gram penalties. Best scores in **bold**, second-best scores underlined.

often **leading to hallucinations**. For example, in one clinical case involving a pregnant woman, the model without a penalty generated a complete and largely accurate hospital course, with minor, mostly harmless errors, such as reporting a discharge followed by a readmission within a few hours. When a bi-gram penalty was applied, the model began producing additional fictional details, including the infant’s sex and weight. Although the bi-gram penalized output received a higher automatic score (+0.02), it was less accurate in reality.

This highlights a **limitation of the *Discharge-Me!* evaluation framework**: outputs containing hallucinations can be favored over fully accurate, non-degenerated text with minor redundancy.

Regardless, from a research perspective, this behavior provides valuable insight: it demonstrates that **GETALP-Mistral7B is capable of incorporating additional clinical details** when such information is available. This suggests that, given complete and accurate input data, the model has the potential to produce more precise and informative outputs.

4.4.3 Redundancy

GETALP-Mistral7B sometimes repeats similar phrases across multiple sentences in the HC, even when adding new and correct information. Although factually accurate, these outputs reveal that the model **struggles** to present information **concisely** and **organize** clinical narratives efficiently. Because new details are added, these repetitions are not classified as degeneration. An example is shown in Figure 4.5.

She was continued on her home dose of potassium. She was continued on her home dose of metformin. She was continued on her home dose of synthroid. She was continued on her home dose of omeprazole.
--

Figure 4.5: Example of repetition in HC. Instead of summarizing medications in a single sentence or list, the model repeats the same phrase for each drug (i.e. *She was continued on her home dose of*).

4.5 Recollection Test

The results of the recollection test are reported in Tables 4.7 and 4.8. For clarity, throughout this section the test set is referred to as *unseen*, while the set overlapping with the training population is referred to as *recollection*. For a detailed description of this experiment, please refer to Section 3.7.5.

To assess statistical significance of potential differences, a Student’s t-test was conducted for each task. Across both tasks, **performance on the *recollection* set is consistently lower** than on the *unseen* set. For the HC task, the AVG score decreases from 0.309 to 0.278 ($p = 0.030$). For the DI task, AVGs are 0.477 and 0.439, respectively ($p = 0.074$). Thus, in HC the difference is statistically significant, while in DI the evidence of a difference is weaker. Importantly, in neither task does the model perform better on *recollection*: average scores are in

Task	Dataset	AVG	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
HC	unseen	0.309	0.127	0.411	0.198	0.281	0.424	0.307	0.351	0.375
	recollection	0.278	0.097	0.378	0.157	0.239	0.398	0.268	0.315	0.374
DI	unseen	0.477	0.279	0.552	0.376	0.453	0.550	0.496	0.543	0.568
	recollection	0.439	0.228	0.525	0.319	0.410	0.523	0.447	0.510	0.552

Table 4.7: Per-task evaluation on *unseen* vs. *recollection* datasets. Best scores in bold.

Rank	Dataset	Overall	BLEU	R-1	R-2	R-L	BERT	METEOR	Align	MEDCON
1	unseen	0.393	0.203	0.482	0.287	0.367	0.487	0.402	0.447	0.471
2	recollection	0.359	0.162	0.452	0.238	0.324	0.460	0.358	0.412	0.463

Table 4.8: Overall scoring system results for the recollection test. Best scores in bold.

fact lower, contrary to the initial hypothesis that exposure to the same patients during training would confer an advantage.

One possible explanation could be a greater concentration of degenerated examples in the *recollection* set (91 vs. 73). However, when degenerated outputs are removed from the evaluation, the difference in average scores persists: 0.347 vs. 0.312 ($p = 0.069$). Although the gap is reduced and the significance weakened, performance remains lower on *recollection*, suggesting that **degeneration alone does not account for the observed effect**.

To further investigate, five non-degenerated `subject_ids` from the *recollection* set were sampled and the corresponding training hospitalizations examined to identify potential overlaps between the generated HC and the training diaries. Upon manual inspection by non-medical experts, **no clear similarities were observed**. This suggests that the observed **lower performance is not clearly linked to the model explicitly recalling** or copying fragments from training data.

Overall, this test shows **no benefits** for GETALP-Mistral7B from having seen the same patients during training. The significant difference in mean scores between the *unseen* and *recollection* sets warrants further investigation, ideally involving clinical experts to interpret potential underlying causes.

Chapter 5

Conclusions

This chapter synthesizes the main findings, design choices, and insights emerging from this study, providing a clear overview of the feasibility and limitations.

Section 5.1 (**Feasibility**) evaluates the practical viability of using GETALP models for generating hospital courses and discharge instructions. It discusses the performance of Mistral7B and Llama3-8B, identifies recurring issues such as degeneration and hallucinations, and highlights differences between HCs and DI in terms of stylistic coherence and reliability.

Section 5.2 (**Source Document Design and Components**) revisits the methodology for converting EHRs into task-specific textual inputs, emphasizing the role of the diary and patient summary in enabling high-quality generation. It also discusses the components of source documents, highlighting the importance of designing EHR interfaces in a doctor-friendly manner.

Section 5.3 (**Investigating Degeneration**) analyzes the causes and impacts of degeneration observed in Mistral7B outputs, including the effect of truncated EOS tokens, prompt length, and model architecture.

Section 5.4 (**Evaluation Framework: A Review**) provides a critical assessment of the scoring system, identifying its limitations in penalizing hallucinations and partially addressing degeneration and redundancy.

Finally, Section 5.5 (**Privacy and Recollection**) addresses patient privacy concerns and the challenges revealed by the recollection test, analyzing potential sources of performance discrepancies and stressing the need for careful evaluation before any public deployment.

5.1 Feasibility

The quality of the hospital courses and discharge instructions generated by the GETALP models demonstrates that **it is feasible to create a clinical large**

language model for automated discharge documentation from electronic health records. Based on the MIMIC-IV dataset, which contains the EHRs of the Beth Israel Deaconess Medical Center, a pipeline for automating discharge summaries sections has been developed. This pipeline first restructures EHRs into textual source documents, which are then used to train and subsequently test GETALP-Mistral7B and GETALP-Llama3-8B. In doing so, the system establishes a new state-of-the-art solution for generating HCs and DI.

Nonetheless, there remain **margins for improvement**. The generation of HCs by GETALP-Mistral7B is affected by issues such as degeneration and redundancy, which are mainly linked to the model’s pre-training or architecture. In contrast, the generation of DI is less prone to such problems, largely due to their stylistic coherence throughout MIMIC-IV. Indeed, DI are almost always structured as a letter to the patient, with standardized opening and closing clauses, whereas the ground-truth HCs display greater variability. They differ in length and style, and, being targeted to a medical audience, contain numerous codes and abbreviations. Some clinicians tend to prefer extended descriptions, while others opt for highly succinct summaries. Even though this problem affects only the Mistral7B version of GETALP, it deserves further analysis, which is provided later in Section 5.3.

Regardless, the models consistently produce **coherent outputs**, even if not always accurate. **Hallucinations**, instances in which the model generates information not present in the source EHRs, can have significant negative consequences in discharge summaries. Mistakes of this kind may lead to delays in follow-ups or misunderstandings, and in the worst case, they could directly compromise patient safety or result in inappropriate interventions. For example, as illustrated in Section 4.4, GETALP-Mistral7B inferred the sex of a newborn infant even though this information was never mentioned in the records.

It can therefore be concluded that creating an automatic pipeline for DS generation is not utopian, even though research should proceed towards **optimizing source documents** and further investigating the technical causes behind Mistral7B’s generative issues.

5.2 Source Document Design and Components

5.2.1 Design

Finding a comprehensive and coherent method to convert EHRs into digestible model inputs was a crucial aspect of this study. EHRs were divided into two broad categories: **structured data** and **clinical notes**. Structured data, being tabular, required a textualization process, whereas clinical notes are already in text format, with the main challenge being what to retain and how to integrate them without excessively increasing the length of the source document.

Two types of source documents were designed, tailored to their respective tasks: the **diary for HCs** and the **patient summary for DI**. Their components are described in detail in Section 3.1, while the generation process is presented in Section 3.2. The design of these documents was guided by the need for **scalability**, as EHRs differ across hospitals and countries. Only essential clinical records, present in most EHRs systems, were integrated, and raw notes were preferred whenever possible, with the goal of enabling a fully automated pipeline for DS generation.

Results reported in Chapter 4 show that **GETALP models outperform state-of-the-art solutions** regardless of the architecture employed. WisPerMed, winners of the *Discharge-Me!* Challenge, provide a useful comparison: they employ similar technological approaches, including Asclepius models trained with QLoRA, but structure the input differently. In their approach, the BERT-score similarity between the target section and other sections is computed, and the original discharge summary is reorganized from the most important section (e.g., HPI, as also observed in this work) to the least important. This ensures that, even in cases of truncation, the most relevant sections appear at the beginning of the input.

This comparison supports the claim that **the superior performance of GETALP models is largely due to the design of the source documents**. Structuring the diary chronologically allows GETALPs to capture the sequence of events, facilitating the completion of facts not explicitly mentioned. Likewise, the conciseness of the patient summaries enables the models to focus on “what” happened, providing patients with clear and accurate instructions for post-discharge care.

5.2.2 Components

As shown in Section 4.2, certain sections of the source documents, such as the **history of present illness**, are **crucial** for producing high-quality outputs. Broadly, clinical notes are more essential than raw structured data. This observation aligns with a logical perspective, as clinical notes are typically more curated and detailed than raw data, but it also introduces challenges for end-to-end discharge summary automation.

Looking forward to a fully automated DS pipeline, it is essential to design an EHR interface that allows clinicians to input critical information in a manner that is both **simple and fast**, ideally co-designed with medical professionals to maximize usability and efficiency. As will be discussed in Section 6, some of this information could be recorded as audio notes and subsequently post-processed by AI systems to generate the corresponding textual document.

5.3 Investigating Degeneration

Although degeneration does not necessarily imply low-quality hospital course, it remains a serious concern. During result analysis, an imperfection in the training process was identified: in some cases, the **EOS token** was truncated. As a consequence, GETALP-Mistral7B was exposed to examples without a proper termination signal, preventing it from fully learning when to stop.

Convinced that this was the main cause of degeneration, a new version of GETALP-Mistral7B was trained after fixing the bug in the training set. After **re-training**, AVG slightly improved from 0.309 to 0.320 (+0.011), and the incidence of degeneration decreased from 29.2% to 26%. However, **degeneration still persisted**.

Beyond the lack of clear closing patterns in the HC section (e.g., “*Follow-ups*” for DI), prompt length appears to play a role. Table 5.1 reports the Pearson correlations between prompt length and AVG of the bug-free version of GETALP-Mistral7B. Degeneration is confirmed to have a negative impact on performance (-0.339 , $p\text{-value} = 3.7 \times 10^{-08}$). More importantly, degeneration correlates positively with prompt length (0.245 , $p\text{-value} = 9.0 \times 10^{-05}$), suggesting that **excessively long inputs are more prone to produce degenerated outputs**. A potential mitigation strategy could involve reducing diary length via summarization before feeding it to the model. To this end, a clinical causal language model might be applied in a zero-shot setting.

At the same time, the correlation analysis shows that degeneration cannot be explained by input length alone. Manual inspection revealed that very short prompts may also degenerate. The **absence of degeneration in GETALP-Llama3-8B**, despite it being trained with the same EOS truncation bug, further underscores that model architecture plays a significant role.

At a first glance, it seems that the EOS bug-free version of GETALP-Mistral7B surpasses GETALP-Llama3-8B. However, this **would not be a fair comparison**, since Llama3-8B would likely benefit as well from the same re-training.

Overall, the degeneration problem remains only **partially understood**, although the main cause of such phenomena can likely be attributed to the generative capabilities of foundation models.

5.4 Evaluation Framework: A Review

The *Discharge-Me!* evaluation framework requires careful reconsideration. As pointed out in Section 4.4, the framework tends to **award higher scores to hallucinated summaries** over incomplete ones. In a clinical context, this behavior is undesirable: it is preferable that the system **refrains** from providing information

Feature	degeneration	prompt_length	AVG
degeneration	1	0.245	-0.339
prompt_length	0.245	1	-0.085
AVG	-0.339	-0.085	1

Table 5.1: Pearson correlation coefficients between degeneration flag, prompt length, and average metric score.

rather than introducing potentially incorrect content. Importantly, none of the metrics currently employed penalizes hallucinations. One possible solution could rely on an **anti-score** based on UMLS clinical concepts, which would reduce the per-task average score according to the number of hallucinations in the generated section. Is important to enhance the support for **engineering new metrics** specifically designed for clinical text generation.

Regardless, **expert review remains essential** in any case. Beyond human judgment, LLMs such as GPT-5 can also be employed for validation, provided that security standards are strictly respected. This includes the possibility of employing services such as Azure OpenAI with human review data opted out, which is fully compliant with the DUA signed with PhysioNet for the use of MIMIC-IV. However, this service is not freely available, and therefore dedicated funding would be required to explore this solution.

5.5 Privacy and Recollection

As detailed in Section 1.2, privacy represents a fundamental concern in clinical environments. Patient health data must be rigorously protected, as it could otherwise be misused for harmful purposes.

The GETALP-Mistral7B recollection test described in Section 4.5 does not provide conclusive results. While it can be stated with confidence that the model does not exploit information from previously seen patients, its performance on the recollection set is lower compared to the unseen set. This indicates that some factor is affecting performance on the recollection set.

Given that MIMIC-IV is a pseudonymized dataset, it is unlikely that the issue arises from memorization of individual patients. Rather, the performance difference is most likely linked to discrepancies in the distribution of clinical cases between the *unseen* and *recollection* sets. In particular, the recollection set may include cases that are more susceptible to report quality issues, such as repetition or other generative artifacts.

In any case, the factors causing GETALP-Mistral7B to underperform on the

recollection test must be thoroughly understood and addressed before any public release of the model.

Chapter 6

Towards an End-to-End Discharge Summary Automation

As discussed at the beginning of this report, realizing an end-to-end AI-assisted system to write discharge summaries could represent a genuine turning point in clinical workflows.

Beyond the automation of hospital courses and discharge instructions, there are other DS sections that can be targeted for automation.

Some categories of clinical data lend themselves more naturally to full automation in both collection and reporting. Medication records, for example, do not necessarily require manual input from physicians: once a drug is prescribed and administered, it can be registered automatically through barcode scanning or electronic dispensing systems. Laboratory results follow a similar pattern. Since they are typically generated by digital analyzers and stored in structured formats, they can be transferred directly to the discharge summary without the need for manual transcription. In these cases, automation is not only feasible but **already widely implemented**, with artificial intelligence playing a potential role mainly in post-processing tasks, such as consolidating duplicate entries, standardizing nomenclature, or filtering out results that are not clinically relevant for the specific report.

Other categories of information inevitably require at least some degree of manual input from healthcare personnel. Data collected during triage, such as findings from the physical examination (e.g., HEENT, cardiovascular, or abdominal assessments), can only be **recorded by the clinician who performed them**. In these cases, the physician or nurse is not expected to engage in a stylistic effort, but rather to accurately report what was observed or what the patient declared. Often,

such entries are standardized through controlled vocabularies or coding systems (e.g., ICD), which means that documentation frequently reduces to selecting the appropriate item from a predefined list. Compared to narrative clinical notes, these records tend to be faster to produce, as their nature lacks a “narrative” dimension and instead relies on structured reporting of objective or categorical findings.

6.1 History of Present Illness, Social Background, Family History

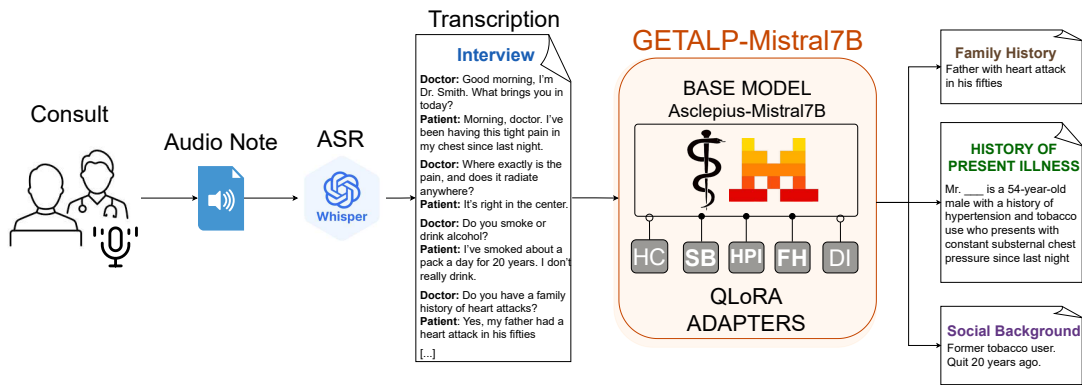


Figure 6.1: Sketch of the automation system for history of present illness (HPI), social background (SB) and family history (FH).

Certain categories of clinical information can only be provided directly by the patient, such as the history of present illness (HPI), family history (FH), and social background (SB). Typically, these sections are generated following a clinician–patient **interview**, after which the physician must manually write the findings into the medical record. A **potential automation pipeline** envisions recording the conversation, applying an automatic speech recognition (ASR) system to transcribe the audio, and then further training three specialized QLoRA adapters to extract and structure the relevant information.

In a preliminary sketch of such a system (Figure 6.1), Whisper (OpenAI) is employed to transcribe the interview, and GETALP-Mistral7B applies the three adapters to generate the HPI, SB, and FH, respectively. Whisper is open source and can be executed offline without relying on OpenAI’s paid API, making it suitable for research or internal hospital deployments.

Obtaining sufficient high-quality clinical interviews to train GETALP-Mistral7B is particularly challenging. While there exist commercial ASR solutions specialized for healthcare—such as Nuance Dragon Medical One, Amazon Transcribe Medical,

and Microsoft Azure Speech for Healthcare—these are generally proprietary and cloud-based. In the open-source domain, no large, publicly available clinical ASR dataset exists to achieve medical-grade performance.

6.2 Patient Medical History

The automation of the patient medical history (PMH) section depends heavily on how EHRs are **organized and managed at the national level**. In countries with centralized healthcare systems, such as France and Italy, it is generally possible to retrieve a patient’s full medical history through a **single access point**. For instance, the Italian *Fascicolo Sanitario Elettronico* consolidates medical records from different providers, making it feasible to automatically populate the PMH section by directly querying national repositories. In such contexts, AI may primarily support tasks of structuring, summarizing, or filtering the retrieved data rather than generating it from scratch.

By contrast, in countries with **privatized** healthcare systems, such as the United States, the fragmentation of records across insurance companies, hospitals, and private clinics complicates seamless integration. Here, accessing a unified longitudinal medical history is significantly more challenging due to policies, interoperability issues, and the proprietary nature of many health records. In such scenarios, the most practical approach is to handle the PMH similarly to sections like HPI, FH, and SB: information must be elicited directly from the patient, recorded in an audio file, and then processed through specialized models.

In conclusion, the strategy for automating this section must be tailored to the structure of the available EHR infrastructure. Where centralized records exist, automation can leverage them directly; in fragmented or privatized systems, patient-provided input remains essential.

6.3 Non-Textual Clinical Documents

Automating clinical documentation also requires addressing non-textual sources of information, such as medical images and physiological signals. In radiology, this challenge is already the subject of active research, exemplified by the RRG24 challenge, which runs in parallel to the *Discharge-Me!* one [11]. The goal of such initiatives is to transform imaging findings into structured or narrative reports, ensuring that clinically relevant details are captured accurately and consistently.

More broadly, medical imaging can be integrated into the documentation pipeline through image **captioning** systems, which analyze the image and generate a descriptive summary. While approaches based on deep learning have shown promise in radiography and computed tomography, some domains remain particularly

challenging. Histopathological images represent a notable example: their extremely high resolution and complex visual features demand specialized preprocessing and domain-specific modeling, making straightforward captioning less feasible.

Other types of non-textual clinical data also play a key role in documentation. For instance, electrocardiograms (ECGs) are widely used and routinely stored in digital formats. Unlike narrative notes, ECG already contain structured information that can be automatically processed. Algorithms for ECG interpretation are well established and can generate preliminary diagnostic statements (e.g., “sinus tachycardia,” “left ventricular hypertrophy”), which may then be reviewed or refined by a physician before inclusion in the discharge summary. Similarly, signals such as electroencephalograms or spirometry results could be integrated using automated analysis pipelines that translate raw measurements into concise, human-readable findings.

In all these cases, **the role of AI is not limited to generation** but extends to harmonization: filtering noise, standardizing terminology, and aligning automatically produced outputs with the structure and requirements of the final discharge summary.

Appendix A

Examples

The aim of this chapter is to provide concrete examples of the documents referenced throughout this report. All examples presented here refer to the same clinical case, corresponding to a single (`subject_id`, `hadm_id`) pair within MIMIC-IV. This consistency allows readers to directly compare the ground truth data with the model predictions. The selected case involves a relatively small number of significant events during hospitalization, which makes it more suitable for visualization

A.1 Prompts

Prompt For Hospital Course Generation

You are an expert clinical assistant.

Your task is to generate a hospital course based on the provided patient diary.

A hospital course refers to the sequence of events that happen to a patient during their stay in a hospital. It includes the patient's initial condition, the treatments they receive, their response to those treatments, and their final condition upon discharge.

Below is the patient's diary, which contains a chronological record of clinical events:

Diary: {diary}

Instructions:

Write the hospital course of the patient. Ensure it includes the patient's initial condition at admission, key treatments and interventions, response to care, and their conditions upon discharge.

Response:

Prompt For Discharge Instructions Generation

You are an expert clinical assistant.

Your task is to generate clear and concise discharge instructions that provide essential guidance for post-hospital care.

Below is a summary of a patient's hospitalization, including their medical, social, and personal history, diagnosis, hospital course, and condition at discharge. Ensure readability for the patient while maintaining medical accuracy.

Patient Summary: {patient_summary}

Instructions:

Write detailed and patient-friendly discharge instructions, including medication guidance, follow-up care, activity restrictions, and warning signs to watch for.

Response:

A.2 Source Documents

MIMIC-IV uses ___ as placeholder for anonymized information. For visualization purposes, some parts have been shortened or omitted.

Patient Summary

Allergies:

Patient recorded as having No Known Allergies to Drugs

History of Present Illness:

Mr. ____ is a ____ year old man who had a mechanical slip and fall in his kitchen. He was taken to the ____ and ____ to have a right ankle fracture with dislocation. His ankle was reduced and he was then transferred to the ____ for further evaluation.

Hospital Course:

Mr. ____ presented to the ____ on ____ via transfer from ____ _____. He was evaluated by the orthopaedic surgery department and found to have a right ankle fracture. He was admitted, consented, and prepped for the operating room. On ____ he was taken to the operating room and underwent an ORIF of his right ankle. He tolerated the procedure well, was extubated, transferred to the recovery room, and then to the floor. On the floor he was seen by physical therapy to improve his strength and mobility. He was cleared by ____ to go home without any need for acute ____.

The rest of his hospital stay was uneventful with his lab data and vital signs within normal limits and his pain controlled. He is being discharged today in stable condition.

Discharge Medications:

1. Oxycodone 5 mg Tablet Sig: ____ Tablets PO Q3H (every 3 hours) as needed for pain: Do not drink, drive, or operate heavy machinery while taking this medication. Disp:*40 Tablet(s)*Refills:*0
2. Enoxaparin 40 mg/0.4 mL Syringe Sig: One (1) 40mg syringe Subcutaneous Q 24H (Every 24 Hours) for 4 weeks. Disp:*28 40mg syringe*Refills:*0
3. Docusate Sodium 100 mg Capsule Sig: One (1) Capsule PO BID (2 times a day). Disp:*60 Capsule(s)*Refills:*2
4. Acetaminophen 325 mg Tablet Sig: ____ Tablets PO Q4H (every 4 hours) as needed.

Discharge Disposition: Home

Discharge Diagnosis: Right ankle dislocation, Right ankle fracture

Discharge Condition: Stable

Diary

— 2132-01-14 —

[17:46-MEDRECON] denies

[21:06-ADMISSION] EW EMER., from EMERGENCY ROOM, insurance: Private.
Died in Hospital?: NO

[21:06-ALLERGIES] No Known Allergies to Drugs

[21:06-CHIEF COMPLAINT] Right ankle fracture

[21:06-HISTORY OF PRESENT ILLNESS] Mr. _____ is a _____ year old man who had a mechanical...

[21:06-PATIENT RECORDS] WHITE M SINGLE 34 y.o. English speaker.

[21:06-PHYSICAL EXAM ADMISSION] Alert and oriented,

Cardiac: Regular rate rhythm,

Chest: Lungs clear bilaterally,

Abdomen: Soft non-tender non-distended,

Extremities: RLE, + sensation/movement, skin intact, + pulses

[22:17-ADMIT] Surgery/Trauma

— 2132-01-16 —

[00:00-PROCEDURE] Open reduction of fracture with internal fixation, tibia and fibula

[08:00-PRESCRIPTION] Enoxaparin Sodium 40mg Syringe SC Q 24H 08

[19:57-CONSULTS] Physical Therapy

[19:57-RESPIRATORY] Oxygen Therapy

[20:00-PRESCRIPTION]: *Bisacodyl* 10mg Suppository PR DAILY:PRN, *Morphine Sulfate* 50mg/50mL Syringe IVPCA ASDIR, *Ondansetron* 2mg/mL-2mL IV Q8H:PRN

[21:35-RADIOLOGY]

STUDY: Right ankle intraoperative study.

HISTORY: Patient with ankle fracture status post ORIF.

FINDINGS: Four fluoroscopic images from the operating room demonstrate <...>

— 2132-01-18 —

[16:00-DISCHARGE CONDITION] Stable

[16:00-DISCHARGE DIAGNOSIS] Right ankle dislocation, Right ankle fracture

[16:00-DISCHARGE MEDICATIONS] *Oxycodone* (5 mg Tablet PO Q3H as needed for pain); *Enoxaparin* (40mg syringe Subcutaneous Q 24H for 4 weeks); *Docusate Sodium* (100 mg Capsule PO BID); *Acetaminophen* (325 mg Tablet PO Q4H as needed).

A.3 Reference and Inference

A.3.1 Hospital Course

Doctor (reference)

Mr. ____ presented to the ____ on ____ via transfer from ____ _____. He was evaluated by the orthopaedic surgery department and found to have a right ankle fracture. He was admitted, consented, and prepped for the operating room. On ____ he was taken to the operating room and underwent an ORIF of his right ankle. He tolerated the procedure well, was extubated, transferred to the recovery room, and then to the floor. On the floor he was seen by physical therapy to improve his strength and mobility. He was cleared by ____ to go home without any need for acute ____.

The rest of his hospital stay was uneventful with his lab data and vital signs within normal limits and his pain controlled. He is being discharged today in stable condition.

AI (inference)

Mr. ____ presented to the ____ on ____ via transfer from ____ with a right ankle dislocation. He was evaluated by the orthopaedic surgery service and found to have a right ankle fracture. He was admitted, consented, and prepped for surgery. On ____ he was taken to the operating room and underwent an ORIF of his right ankle fractures. He tolerated the procedure well, was extubated, transferred to the recovery room, and then to the floor. On the floor he was seen by physical therapy to improve his strength and mobility.

The rest of his hospital stay was uneventful with his lab data and vital signs within normal limits and his pain controlled. He is being discharged today in stable condition.

A.3.2 Discharge Instructions

Doctor (reference)

Continue to be touchdown weight bearing on your right leg.
Continue your lovenox injections for a total of 4 weeks after surgery.
If you notice any increased drainage, swelling, or if you have any pain not relieved by rest of pain medication, or if you have a temperature greater than 101.5 please call the office or come to the emergency department.

Physical Therapy:

Activity: Activity as tolerated

Right lower extremity: Non weight bearing

Treatments Frequency: Keep splint clean and dry

Do NOT put anything down your splint

Followup Instructions:

AI (inference)

Continue to be non-weight bearing on your right leg.
Continue your lovenox injections as instructed.
Please take all your medication as prescribed.
If you have any increased redness, drainage, or swelling, or if you have a temperature greater than 101.5, please call the office or come to the emergency department.

Physical Therapy:

Activity: Activity as tolerated

Right lower extremity: Non weight bearing

Treatments Frequency: Staples/sutures out 14 days after surgery

Dry sterile dressing daily or as needed for drainage or comfort

Followup Instructions:

Appendix B

MIMIC-IV: Statistics

This chapter is intended to provide statistical information about MIMIC-IV, with a particular focus on the number of encounters and the length of textual documents, as these are the primary factors contributing to token limit overflow.

B.1 Number of events per admission

The statistics and data reported here refers to EHRs after MIMIC-IV pre-processing, described in Sections 3.4.2, 3.4.3 and 3.4.1, on the training split obtained as explained in 3.4.7.

Events are defined on MIMIC-IV log tables, created through the LogBuilder described at 3.2. Given a hospitalization admission identifier `hadm_id`, an event is represented by a `timestamp-action` pair. Therefore, the number of unique `timestamp-action` pairs corresponds to the number of events recorded for that admission.

For each hospitalization, there are on average 79.36 events. The majority of events come from tables that record prescriptions (*prescriptions*), laboratory tests (*labevents*), and documentation (*chunk_ds*, *poe*). Some actions are consistently present for all admissions, such as general admission (*admit*) and patient record (*patient_records*) entries, each appearing exactly once per admission. Other tables, such as *medrecon* and *microbiologyevents*, are present only in a subset of admissions.

Table B.1 summarizes the statistics for each table considered, including minimum, median, average, maximum, variance, and total number of admissions with at least one event.

Table	min	median	mean	max	var	count
admit	1	1	1.00	1	0.00	116694
chunk_ds	1	11	10.86	12	1.18	116544
disch	1	1	1.00	1	0.00	99467
drgcodes	1	2	1.92	2	0.07	99434
labevents	1	5	11.10	1352	431.76	105721
medrecon	1	7	7.87	67	32.92	41325
microbiologyevents	1	1	2.35	64	6.75	45701
patient_records	1	1	1.00	1	0.00	116694
poe	1	5	9.47	790	221.22	109346
prescriptions	1	22	33.38	1447	1266.33	116383
procedures_icd	1	2	3.08	40	8.21	70137
radiology	1	3	4.29	221	34.25	92641
transfers	1	2	2.86	20	2.26	116693

Table B.1: Summary statistics of events per admission in MIMIC-IV tables after preprocessing and on the training split. Statistics include minimum, median, average, maximum, variance, and total number of admissions.

B.2 Text Length

This section reports statistics on the textual length of clinical documents, presented via histograms, boxplots, and summary tables.

As discussed throughout this report, the length of clinical text can pose significant challenges when working with LLMs. In particular, the token context of GETALP-Mistral7B has been extended from the original $8k$ tokens, on which Mistral7B was trained, to $15k$ tokens using the Unsloth framework. While the model can still operate beyond $15k$ tokens, its performance is not guaranteed. For this reason, tables in this appendix include the probability of encountering documents exceeding both $8k$ and $15k$ tokens.

Tables B.2, B.4, and B.3 summarize word counts for different sources of clinical text: general log messages from MIMIC-IV-HOSP and MIMIC-IV-ED, clinical notes extracted from MIMIC-IV DS, and POE messages, respectively. All statistics are computed on the training split, constructed from the MIMIC-IV admission table as outlined in Section 3.4.7. Word boundaries were defined by blank spaces, newlines, tabs, and punctuation, using a rule-based approach. Importantly, word counts are computed **before** combining messages with the same **timestamp-action** into a single event, allowing an assessment of the informational content of individual communications. Figures B.1 and B.2 illustrate the distributions of these word counts through histograms and boxplots.

Although word counts are less precise than token counts, they provide a useful approximation of text length independent of the tokenizer used later. Token counts were then computed for source documents, ground truths of target sections, and the resulting training prompt, which combines system message, source document, model instruction, and ground truth. Tables B.5-B.6 summarize token counts for hospital courses and discharge instructions across training, test, and recollection splits. Figures B.7 and B.8 visualize the distribution of token counts for the resulting training prompts.

By comparing word count statistics (Figure B.1) with token count statistics (Tables B.5 and B.6), a rough word:token ratio of 1 : 2 emerges. Therefore, doubling the word count values in Tables B.2, B.4, and B.3 gives a reasonable estimate of the token equivalent.

Finally, Table B.7 reports the statistics for the training prompt, combining system message, source document, model instruction, and ground truth. Figures B.7 and B.8 show the corresponding histograms and boxplots.

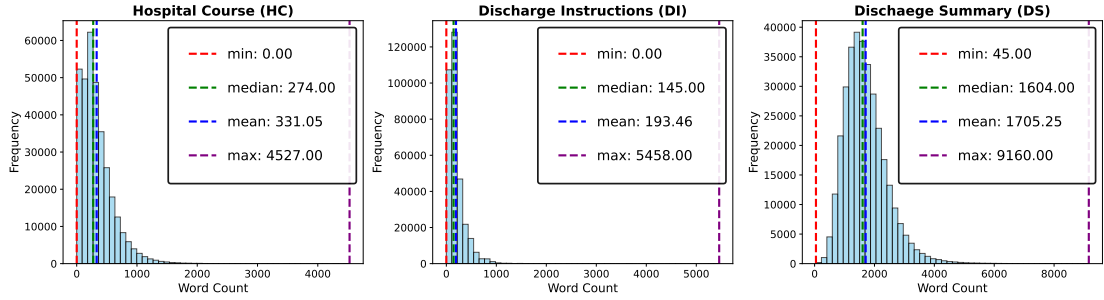


Figure B.1: Word Count - Histogram of wordcount counts for MIMIC-IV’s hospital course, discharge instructions and discharge summary.

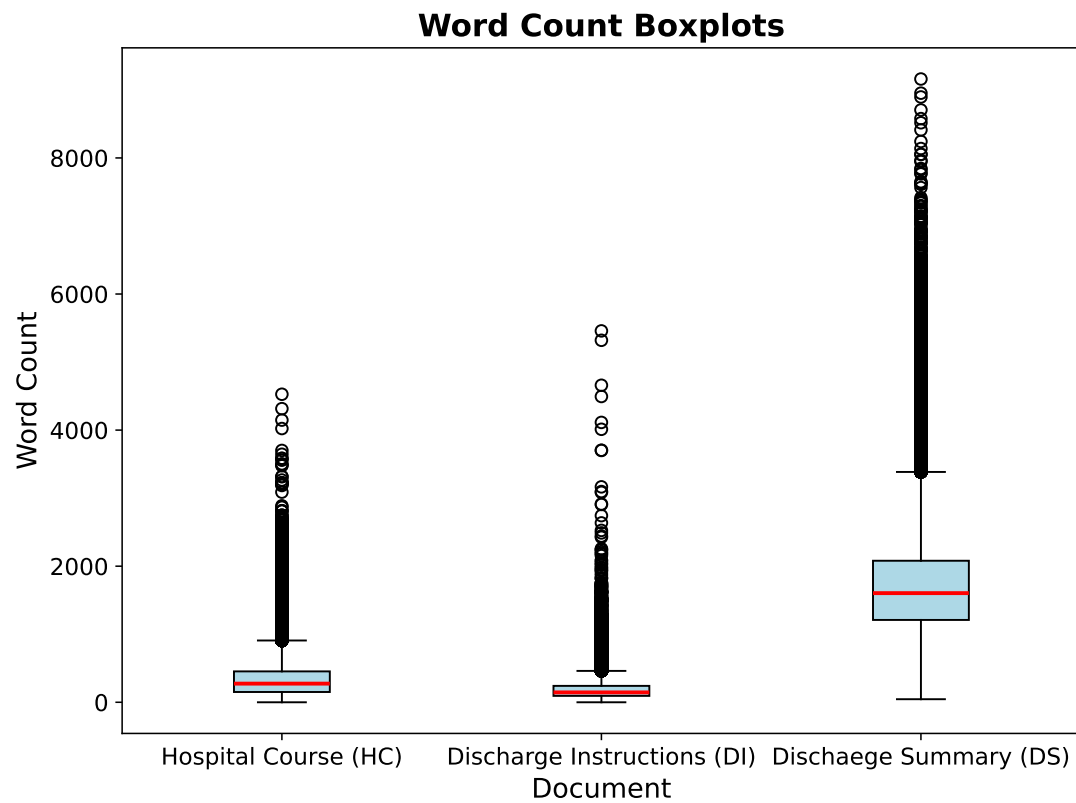


Figure B.2: Word Count - Histogram of wordcount counts for MIMIC-IV's hospital course, discharge instructions and discharge summary.

Event	min	max	mean	median	var	count
Admission	10	20	12.5	12	1.46	116,694
Discharge	1	4	1.81	1	0.99	99,467
DRG Code	1	18	6.31	6	8.08	191,063
LAB-abnormal	2	321	15.2	9	311.23	684,014
LAB-resolved	2	88	8.41	6	36.7	489,723
Medrecon	1	20	1.70	1	1.53	325,081
Microbiology	3	206	19.07	11	418	107,368
Patient Records	6	13	8.28	8	0.48	116,694
Prescription	1	38	8.33	8	4.90	3,884,499
Procedure	1	19	6.76	6	9.33	215,828
Transfer-ED	2	2	2	2	0.00	79,725
Transfer-transfer	1	7	1.87	1	2.34	136,986
Transfer-admit	1	7	1.81	1	1.81	116,677
Radiology	1	2,911	168	107	24,742	397,807

Table B.2: Word count of log messages for MIMIC-IV events. The Procedures table contains only brief specifications on the type of intervention, not detailed clinical notes, which could lead to a significant increase in diary length in other EHR systems.

Event	min	max	mean	median	var	count
Cardiology	1	3	1.05	1	0.07	100,207
Consults	1	13	3.25	2	6.60	218,806
Critical Care	2	3	2.14	2	0.12	13,679
General Care	1	12	5.34	6	15.54	476,787
Hemodialysis	1	1	1.00	1	0.00	7,689
IV therapy	2	9	7.50	9	8.14	130
Neurology	1	2	1.00	1	0.00	11,637
OB	1	2	1.65	2	0.23	4,130
Respiratory	1	3	1.97	2	0.13	184,865
TPN	1	3	2.02	2	0.10	17,249

Table B.3: Word count of POE messages in MIMIC-IV. The POE table records only brief notes on the type of exam or intervention. This explains the short average message length; EHR systems with more detailed notes could significantly impact the diary length. Acronyms: OB = Obstetrics, TPN = Total Parenteral Nutrition.

Note	min	max	mean	median	var	count
Allergies	1	1,182	6	7	29	116,541
Chief Complaint	0	1,128	3	3	136	112,929
Discharge Condition	0	1,188	16	15	352	116,066
Discharge Diagnosis	0	337	11	7	146	112,148
Discharge Medications	0	1,114	132	117	7,450	112,129
Family History	0	1,882	15	8	1,216	109,614
HPI	0	2357	230	206	26,516	114,220
MSIP	0	839	9	4	450	85,418
Past Medical History	0	4,094	41	22	4,973	109,394
Physical Exam Admission	0	1,545	103	88	6,898	1081,94
Physical Exam Discharge	0	1,568	74	70	2,638	56,405
Social History	1	1,721	4.38	1	640	112,836

Table B.4: Word count of clinical notes extracted from DS in MIMIC-IV. The count was performed before removing empty sections. Acronyms: HPI: History of Present Illness, MSIP: Major Surgical or Invasive Procedure

	min	25%	50%	95%	max	mean	p>8K	p>15K
Train								
Diary	28	2,331	3,324	11,426	32,768	4,507	0.11	0.03
HC	3	332	520	1632	8,222	665	0	0
Prompt	178	2,481	3,474	11,576	32,768	4,657	0.13	0.03
Test								
Diary	28	2,328	3,322	11,317	32,768	4,505	0.11	0.03
HC	2	332	524	1624	6441	667	0	0
Prompt	178	2,478	3,472	11,467	32,768	4,654	0.11	0.03
Test Recollection								
Diary	28	2,809	3,815	11,324	32,768	4,859	0.11	0.03
HC	2	398	656	1,763	7,773	774	0	0
Prompt	178	2,959	3,965	11,474	32,768	5,009	0.12	0.03

Table B.5: Hospital Course - Token Count. Acronyms: HC = Hospital Course

	min	25%	50%	95%	max	mean	p>8K	p>15K
Train								
PS	45	974	1,367	2,982	9,198	1,529	$11 \cdot 10^{-5}$	0
DI	2	164	253	941	8,946	345	0	0
Prompt	166	1,095	1,488	3,103	9,319	1650	$11 \cdot 10^{-5}$	0
Test								
PS	83	976	1,364	2,999	9,752	1,531	$3.5 \cdot 10^{-5}$	0
DI	7	163	251	935	5,152	343	0	0
Prompt	204	1,097	1,485	3,120	9,873	1,652	$3.5 \cdot 10^{-5}$	0
Test Recollection								
PS	99	1,134	1,598	3,214	9,098	1,736	$8.2 \cdot 10^{-5}$	0
DI	7	161	244	863	8,143	322	0	0
Prompt	123	1,255	1,719	3,335	9,219	1,857	$8.2 \cdot 10^{-5}$	0

Table B.6: Discharge Instructions - Token Count. Acronyms: PS = Patient Summary, DI = Discharge Instructions.

	DI	HC
min	173	280
25%	1440	2911
50%	1865	4099
95%	3495	12767.3
max	10447	32768
mean	1994.31	5312.38
p>8000	0.0001	0.0144
p>15000	0	0.0035

Table B.7: Training Prompt - Token Count. Acronyms: DI = Discharge Instructions, HC = Hospital Course

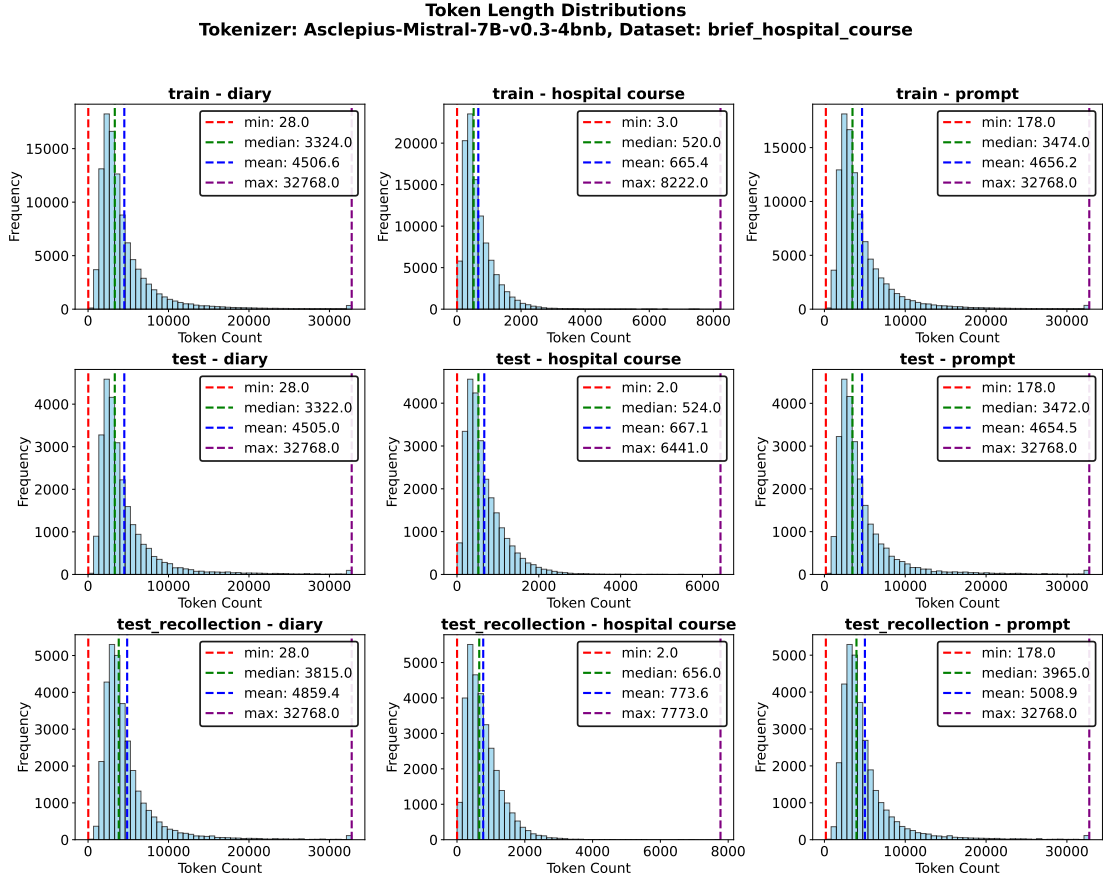


Figure B.3: Hospital Course - Histograms of token counts per split.

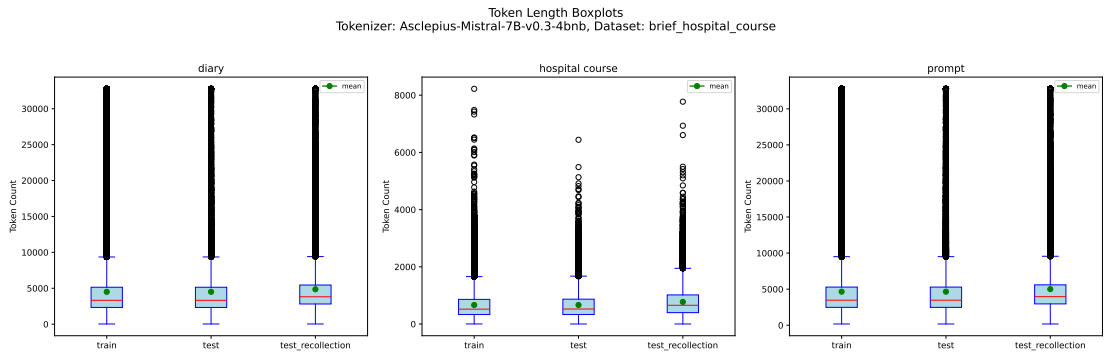


Figure B.4: Hospital Course - Boxplot of token counts per split.

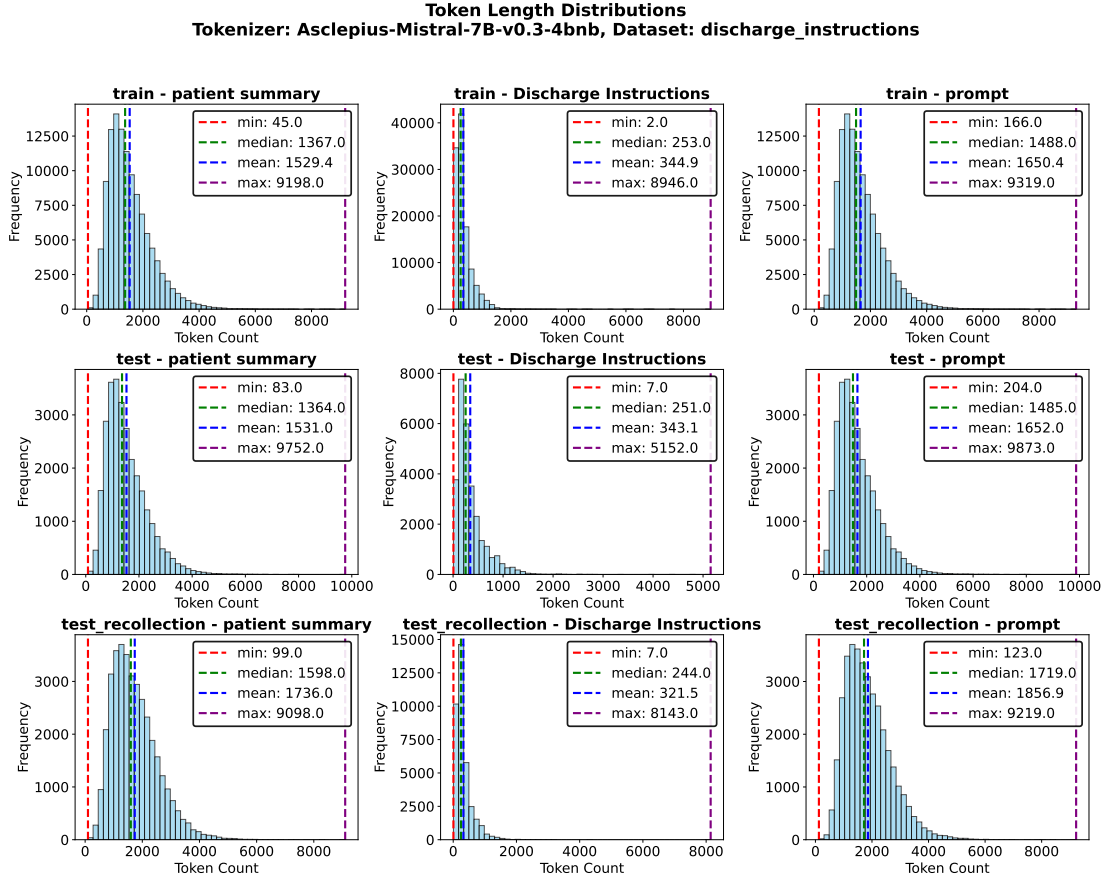


Figure B.5: Discharge Instructions - Histograms of token counts per split.

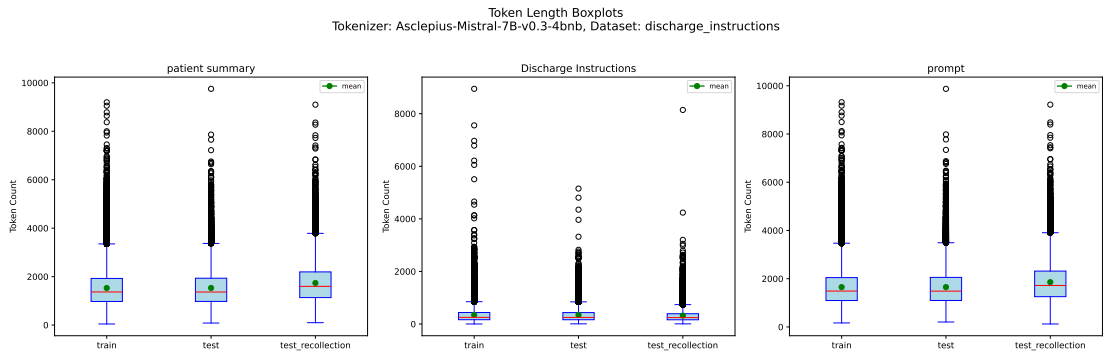


Figure B.6: Discharge Instructions - Boxplot of token counts.

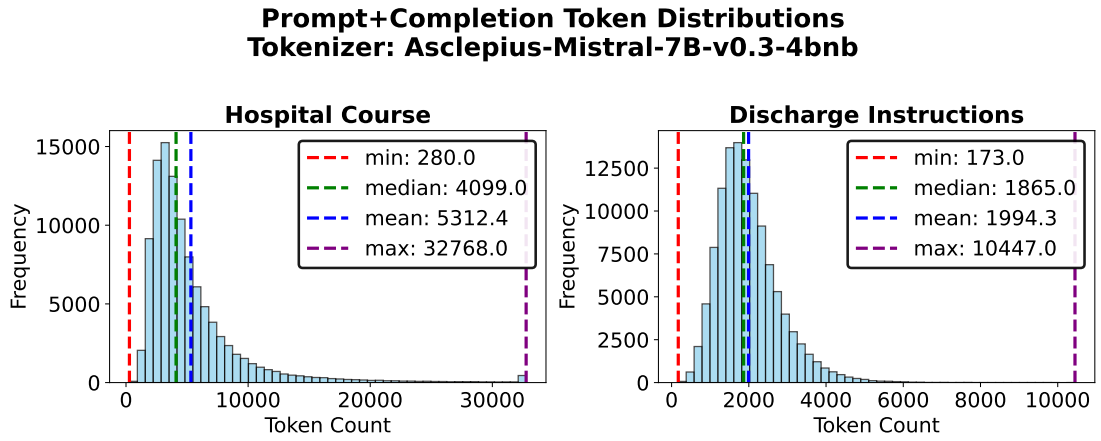


Figure B.7: Training Prompt - Histogram of token counts for Hospital Course and Discharge Instructions.

Tokenizer: Asclepius-Mistral-7B-v0.3-4bnb

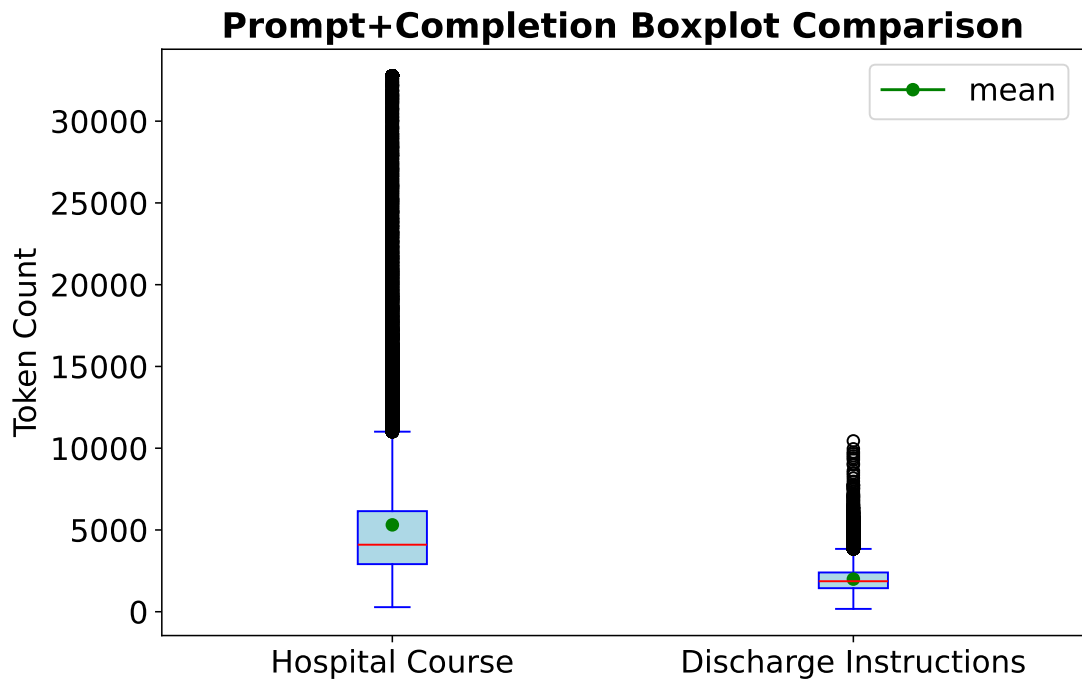


Figure B.8: Training Prompt - Boxplot of token counts for Hospital Course and Discharge Instructions.

Appendix C

Technical Details

This appendix provides a detailed overview of the hardware, software, and carbon footprint associated with the training, evaluation, and inference phases of the project.

C.1 Model Architectures

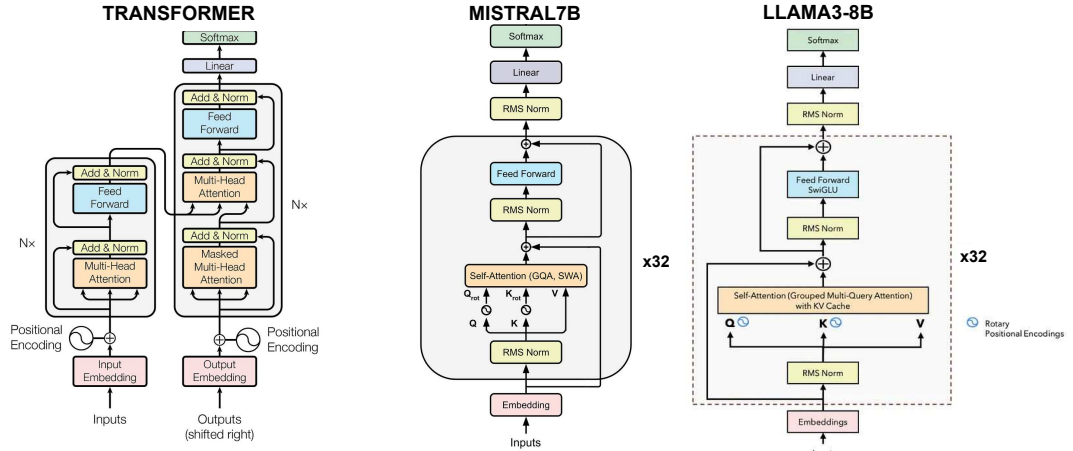


Figure C.1: Architectures - Transformer vs Mistral7B vs Llama3-8B

Two transformer-based architectures were employed in this project: Llama3-8B and Mistral7B. Both models belong to the class of decoder-only, causal language models, featuring multi-head self-attention and feed-forward layers arranged in a stack of transformer blocks. The Llama3-8B architecture comprises 32 transformer layers, with 64 attention heads and an embedding size of 8192. The Mistral7B

model is composed of 32 transformer layers, with 48 attention heads and a hidden dimension of 4096. Both models apply rotary positional embeddings (RoPE) and utilize sparsity-inducing optimizations such as grouped-query attention (GQA) to reduce inference latency. Additionally, **Mistral7B** employs Sliding Window Attention, which allows the model to efficiently handle longer contexts by limiting attention computation to a moving window of tokens.

The **Llama3-8B** model has approximately 8.0 billion parameters, with a storage footprint of ~ 16 GB in 16-bit precision, and was originally trained with a context length of 8192 tokens. The **Mistral7B** model has approximately 7.3 billion parameters, with a storage footprint of ~ 14 GB in 16-bit precision, and was originally trained with a context length of 4096 tokens.

To explore further details, we remind to the works whom introduced Llama3-8B [107] and Mistral7B [108].

C.2 Training Details

C.2.1 Parameters

Parameter	Mistral7B	Llama3
Rank (r)	16	16
α	16	16
Dropout	0.0	0.0
Bias	none	none
Target modules	q,k,v,o,gate,up,down	q,k,v,o,gate,up,down
Trainable params per task	41.9M (0.60%)	41.9M (0.52%)

Table C.1: QLoRA configuration.

The configuration draw inspiration from Damm et al.[96], which trains Asclepius models through QLoRA as of this work. Both Asclepius-Llama3-8B and Asclepius-Mistral-7B-v0.3 were fine-tuned under the same setup. LoRA adapters were applied on projection layers and MLP components, following the configuration reported in Table C.1. For each architecture, two LoRA adapters were trained, one dedicated to HC and the other to DI. Each adapter adds 41.9M trainable parameters, yielding a total of 83.8M trained parameters per model. At inference time, however, only the adapter corresponding to the task is loaded, so the effective size of the model is its base plus 41.9M parameters.

Training arguments were kept consistent across tasks and are summarized in Table C.2. For reproducibility, training was performed with a fixed random seed

Argument	Mistral7B	Llama3
Batch Size (BS)	4	8
Gradient Accumulation (GA)	4	2
Effective Batch Size (EBS)	16	16
Learning Rate (LR)	2e-4	2e-4
Warmup steps	5	5
Epochs	1	1
Optimizer	AdamW 8-bit	AdamW 8-bit
Weight decay	0.01	0.01
Scheduler	Linear	Linear

Table C.2: Training arguments.

(3407).

C.2.2 Memory Usage and Learning Curves

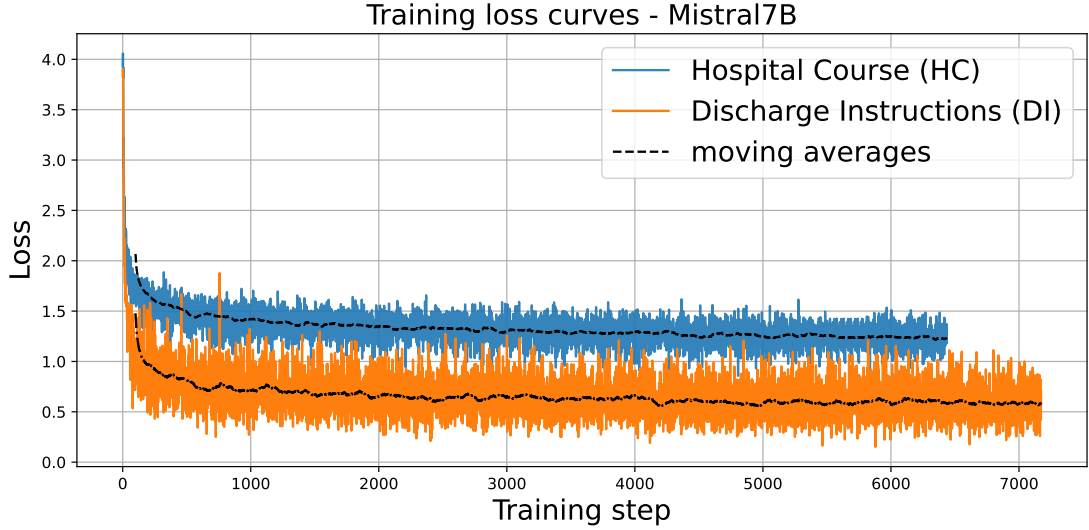


Figure C.2: Learning curves for GETALP-Mistral7B. Both tasks reach a plateau around step 2,000, although a slight downward trend remains visible until the end of training. The model stabilizes at lower loss values for DI (min = 0.151, moving average min = 0.558) compared to HC (min = 0.815, moving average min = 1.214). This indicates better adaptation on the DI task, which is further confirmed by downstream evaluation, while suggesting margins for improvement remain on HC.

Training was performed through QLoRA with 4-bit weight loading, while gradients and optimizer states were stored in bfloat16. Unsloth reserved 4.051 GB of vRAM during training. Datasets comprised 104,528 examples for HC generation and 114,775 examples for DI generation, yielding 6,533 and 7,173 training steps respectively. Both loss curves stabilize after approximately 2,000 steps, with a residual decreasing trend until completion. DI exhibits lower final loss values, indicating more efficient adaptation compared to HC.

C.3 Inference Details

Parameter	Value
Temperature	1
Maximum new tokens	1000
Maximum sequence length	15,000
Precision	4-bit loading, bfloat16 for computations

Table C.3: Inference parameters.

Inference was performed with a maximum sequence length of 15,000 tokens and a temperature of 1.0, generating up to 1,000 new tokens per prompt.

Models were loaded in 4-bit precision, meaning that each parameter is stored using only 4 bits (half a byte). During forward computation, each quantized parameter q is expanded to bfloat16 using the scale and zero-point of its quantization block:

$$x^{\text{float}} = \text{scale}_i \cdot q + \text{zero_point}_i, \quad q \in \{0, 1, \dots, 15\} \quad (\text{C.1})$$

$$\text{scale}_i = \frac{\max(x_i) - \min(x_i)}{15}, \quad \text{zero_point}_i = \min(x_i)$$

where i indexes the quantization block the parameter belongs to. The quantization block comprises typically a small group of weights.

Thanks to 4-bit quantization, the model scale is only 1.77 GB, while loading the same model in bfloat16 would require approximately 5.53 GB.

C.4 Hardware

The training phase was carried out exclusively on a machine equipped with an AMD EPYC 9354 32-Core Processor (128 logical cores), 1,007.56 GB of system RAM, and a NVIDIA H100 NVL GPU with 94 GB of vRAM. The computational

power provided by this setup was essential to perform the intensive gradient and optimizer calculations required during training. During this phase, the NVIDIA H100 NVL GPU operated at an average power consumption of 340 W, reflecting the high demand of the training process.

For inference, tasks were executed on one of two machines, as decided by the OAR scheduler, depending on resource availability. The first machine, hosting the NVIDIA H100 NVL GPU, was used when the resources were available. The second machine, equipped with an Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz and a NVIDIA Quadro RTX 8000 GPU, was used at other times. In both cases, the GPU operated at a reduced power consumption of 195 W during inference, as the workload is less demanding, with no gradient or moment calculations involved.

Preprocessing tasks were performed entirely on CPU, utilizing multithreading as needed.

All machines were located in Pays de la Loire, France.

C.5 Software

All scripts were written in Python (version 3.10.16), and datasets and models were privately hosted on Hugging Face. Version control was managed through a private GitHub repository. Training and inference were supported by the Unsloth framework, which enables memory-efficient model training and provides mechanisms to extend the context window. Due to the use of the MIMIC-IV dataset, subject to PhysioNet confidentiality policies, models, datasets, and the source code repository cannot be made publicly available.

C.6 Environment

The carbon footprint was calculated using CodeCarbon (v. 2.8.3), with updated carbon intensity values retrieved from the website Electricity Maps to reflect the specific conditions at the time of training and inference. The total emissions for each phase were computed by multiplying the emission rate by the total duration of the phase, expressed in seconds. Training activities were mostly conducted between March and April 2025.

C.6.1 Training Phase

The training phase was characterized by low carbon emissions due to France’s energy profile. Training durations and associated emissions are summarized in Table C.4.

Model	Period	Duration	CI (g/kWh)	ER (mg/s)	CF (kg)
M7B-HC	Apr. 5-6	38:13:09	22.5	5.05	0.70
M7B-DI	Apr. 6-7	12:23:41	22.5	4.69	0.21
L8B-HC	Mar. 26-27	29:55:44	37.5	7.96	0.86
L8B-DI	Mar. 23-24	10:47:34	39	7.81	0.30

Table C.4: Training carbon emissions per model. Duration is reported in the format hh:mm:ss. Acronyms: CI: Carbon Intensity, ER: Emission Rate, CF: Carbon Footprint, M7B: Mistral7B, L8B: LLama3-8B, w/o R: without Radiology reports

The total carbon footprint for the training phase amounts to 2.07 kgCO₂eq, which corresponds approximately to driving 8 km with an average gasoline car [109].

The GETALP-Mistral7B EOS bug-free training, performed between Sep. 27-28, was not included in Table C.4 since it occurred after the main experimental phase. Adding this final training to the previous calculations increases the total carbon footprint to **2.79** kgCO₂eq.

C.6.2 Inference Phase

Inference was performed between April 17 and April 27, 2025. Due to the absence of gradient calculations, the computational load was reduced, resulting in lower GPU power consumption. An average carbon intensity peak of 26 gCO₂eq/kWh was recorded during this period. The total inference time across all models, including ablation studies, amounted to 8 hours, 19 minutes, and 43 seconds, leading to overall emissions estimated to be under 0.1 kgCO₂eq.

C.6.3 Total Emissions

By summing the emissions from both training and inference, the total carbon footprint of the project amounts to approximately 2.89 kgCO₂eq. This notably low impact is primarily due to France’s energy grid, which relies heavily on nuclear energy, ensuring one of the lowest carbon intensities in the world.

Appendix D

Metrics

This chapter presents the mathematical details of the eight metrics employed in this work: BLEU, ROUGE-1, ROUGE-2, ROUGE-L, METEOR, BERTScore, AlignScore, and MEDCON [37, 38, 39, 40, 41, 42].

As discussed in Section 3.6, these metrics are typically defined in their generalized multi-reference formulation. In the present study, however, only a single reference is available for each encounter, resulting in simplified computations compared to the general case.

This chapter is not intended to replace the original publications. For comprehensive explanations and theoretical background, readers are referred to the cited sources.

The chapter begins by introducing general definitions that are shared across multiple metrics, followed by a detailed mathematical formulation of each individual metric.

D.1 Candidate, Reference, n -grams, counting, F1-score

Let $\mathcal{C}$ and $\mathcal{R}$ be the candidate and reference text, respectively. The length of a text is defined as the number of words within the text, denoted by $|\cdot|$.

Let g be an n -gram, namely a sequence of n words. Let C and R be the two sets of n -grams obtained by applying the n -gram split to $\mathcal{C}$ and $\mathcal{R}$. The number of n -grams in a set defines its cardinality, indicated as $|C|$ or $|R|$.

The counting function $\text{Count}(g, S)$ counts how many times g appears in a set of n -grams S . The match count of g between C and R is defined as

$$\text{Count}_{\text{match}}(g, C, R) = \min \left(\text{Count}(g, C), \text{Count}(g, R) \right)$$

The F1-score is the harmonic mean between precision and recall. Given precision p and recall r , it is calculated as:

$$F1 = \frac{2 \cdot p \cdot r}{p + r}$$

D.2 BLEU

See D.1 for definitions.

The modified n -gram precision p_n of a candidate text $\mathcal{C}$ with respect to a reference text $\mathcal{R}$ is calculated as follows:

$$p_n = \frac{\sum_{g \in \mathcal{C}} \text{Count}_{\text{clip}}(g, \mathcal{C}, \mathcal{R})}{|\mathcal{C}|}$$

The Brevity Penalty (BP) penalizes candidate texts that are shorter than the reference. It is computed as:

$$BP = \begin{cases} 1 & \text{if } |\mathcal{C}| > |\mathcal{R}| \\ e^{(1 - \frac{|\mathcal{R}|}{|\mathcal{C}|})} & \text{if } |\mathcal{C}| \leq |\mathcal{R}| \end{cases}$$

Finally, the BLEU- N score is given by the geometric mean of modified n -gram precisions p_n from 1 to N , scaled by the brevity penalty BP :

$$BLEU-N = BP \cdot \left(\prod_{n=1}^N p_n \right)^{\frac{1}{N}}$$

D.3 ROUGE

See D.1 for definitions.

D.3.1 ROUGE-N

ROUGE- N measures the overlap of n -grams between candidate and reference. The recall r and precision p are defined as:

$$r = \frac{\sum_{g \in \mathcal{R}} \text{Count}_{\text{clip}}(g, \mathcal{C}, \mathcal{R})}{\sum_{g \in \mathcal{R}} \text{Count}(g, \mathcal{R})}, \quad p = \frac{\sum_{g \in \mathcal{C}} \text{Count}_{\text{clip}}(g, \mathcal{C}, \mathcal{R})}{\sum_{g \in \mathcal{C}} \text{Count}(g, \mathcal{C})}$$

where $n = 1$ for ROUGE-1 and $n = 2$ for ROUGE-2.

Finally, ROUGE- N is given by the harmonic mean of p and r (F1-macro D.1).

D.3.2 ROUGE-L

ROUGE-L evaluates the overlap between a candidate text $\mathcal{C}$ and a reference text $\mathcal{R}$ by considering the **Longest Common Subsequence (LCS)**, which is the longest sequence of words that appears in both texts in the same order, even if the words are not consecutive. This allows ROUGE-L to capture sentence-level structure beyond exact n -gram matches.

Formally, let $\text{LCS}(\mathcal{C}, \mathcal{R})$ denote the length of the LCS between $\mathcal{C}$ and $\mathcal{R}$. Then the recall r , precision p are defined as:

$$r = \frac{\text{LCS}(\mathcal{C}, \mathcal{R})}{|\mathcal{R}|}, \quad p = \frac{\text{LCS}(\mathcal{C}, \mathcal{R})}{|\mathcal{C}|}$$

Example: Consider the reference and candidate sentences:

- Reference: “the cat is on the wooden table”
- Candidate: “the cat sat on the mat”

The LCS between these sentences is “the cat on the”, preserving the word order while ignoring mismatched words. Its length is 4, while the reference has 7 words and the candidate has 6 words.

$$r = \frac{4}{7}, \quad p = \frac{4}{6}, \quad F1 = \frac{2 \cdot \frac{4}{7} \cdot \frac{4}{6}}{\frac{4}{7} + \frac{4}{6}} \approx 0.615$$

D.4 METEOR

See D.1 for basic definitions.

The METEOR score between a candidate $\mathcal{C}$ and a reference $\mathcal{R}$ is computed through the following steps:

1. **Alignment.** Find the set of unigram matches between $\mathcal{C}$ and $\mathcal{R}$, considering exact matches, stem matches, synonym matches, or paraphrase matches.
2. **Precision and recall.** Let m be the number of matched unigrams, then:

$$p = \frac{m}{|\mathcal{C}|}, \quad r = \frac{m}{|\mathcal{R}|}$$

3. **Harmonic mean with recall weight.** The score F_{mean} is a harmonic mean of p and r , giving more weight to recall:

$$F_{mean} = \frac{10 \cdot p \cdot r}{r + 9p}$$

4. **Fragmentation penalty.** Let ch be the number of contiguous matched chunks, then the penalty is:

$$Penalty = 0.5 \cdot \left(\frac{ch}{m}\right)^3$$

5. **Final score.**

$$METEOR = (1 - Penalty) \cdot F_{mean}$$

This formulation makes METEOR more sensitive to both adequacy (recall, synonymy, paraphrasing) and fluency (ordering, chunk fragmentation) than purely n -gram-based metrics.

D.5 BERTScore

Let $\mathbb{C} = \{c_1, \dots, c_{|\mathbb{C}|}\}$ and $\mathbb{R} = \{r_1, \dots, r_{|\mathbb{R}|}\}$ denote the candidate and reference token sequences, respectively. BERTScore computes contextual embeddings $\mathbf{e}_i^c$ and $\mathbf{e}_j^r$ for each token $c_i \in \mathbb{C}$ and $r_j \in \mathbb{R}$ using a pretrained transformer model.

The similarity between tokens is given by the cosine of their embedding vectors:

$$s_{ij} = \frac{\mathbf{e}_i^c \cdot \mathbf{e}_j^r}{\|\mathbf{e}_i^c\| \|\mathbf{e}_j^r\|}$$

Precision and recall are obtained by selecting, for each token, the maximum similarity with tokens in the opposite sequence:

$$\text{Precision} = \frac{1}{|\mathbb{C}|} \sum_{i=1}^{|\mathbb{C}|} \max_{j \in \mathbb{R}} s_{ij}, \quad \text{Recall} = \frac{1}{|\mathbb{R}|} \sum_{j=1}^{|\mathbb{R}|} \max_{i \in \mathbb{C}} s_{ij},$$

The final BERTScore is computed as the harmonic mean of precision and recall (F1-score).

D.6 AlignScore

For the computation of AlignScore, the candidate text $\mathcal{C}$ is split into sentences, and the reference text $\mathcal{R}$ is split into chunks (each containing roughly 350 tokens). Let $\mathbb{C} = \{c_1, \dots, c_m\}$ be the set of candidate sentences and $\mathbb{R} = \{r_1, \dots, r_n\}$ be the set of reference chunks.

For each pair (c_i, r_j) , the NLI model produces logits over three classes: *entailed*, *contradicted*, and *neutral*:

$$\mathbf{z}_{ij} = [z_{\text{entail}}(c_i, r_j), z_{\text{contradict}}(c_i, r_j), z_{\text{neutral}}(c_i, r_j)]$$

The probability that c_i is entailed by r_j is obtained via the softmax over these logits:

$$p_{\text{entail}}(c_i, r_j) = \frac{\exp(z_{\text{entail}}(c_i, r_j))}{\exp(z_{\text{entail}}(c_i, r_j)) + \exp(z_{\text{contradict}}(c_i, r_j)) + \exp(z_{\text{neutral}}(c_i, r_j))}$$

This produces an $m \times n$ matrix of entailment probabilities. For each candidate sentence c_i , we take the maximum probability across all reference chunks:

$$s_i = \max_{j \in \{1, \dots, n\}} p_{\text{entail}}(c_i, r_j)$$

Finally, the AlignScore is obtained by averaging over all candidate sentences:

$$\text{AlignScore}(\mathcal{C}, \mathcal{R}) = \frac{1}{m} \sum_{i=1}^m s_i$$

D.7 MEDCON

Let $\mathbb{C}$ and $\mathbb{R}$ be the sets of UMLS concepts extracted respectively from the candidate and the reference text, restricted to the selected semantic groups.

The precision p and recall r are defined as:

$$p = \frac{|\mathbb{C} \cap \mathbb{R}|}{|\mathbb{C}|}, \quad r = \frac{|\mathbb{C} \cap \mathbb{R}|}{|\mathbb{R}|}$$

The final MEDCON score is computed as the harmonic mean of precision and recall (F1-score).

Bibliography

- [1] Epic Systems Corporation. *Epic / Software for Healthcare*. 2025. URL: <https://www.epic.com/> (visited on 08/11/2025) (cit. on pp. i, 9).
- [2] *Epic EHR vs. Cerner EHR: A Comprehensive Comparison*. 2025. URL: <https://www.ehrinpractice.com/epic-ehr-vs-cerner-ehr-comparison.html> (visited on 08/11/2025) (cit. on pp. i, 9).
- [3] Oracle Corporation. *Oracle Health*. 2025. URL: <https://www.oracle.com/health/> (visited on 08/11/2025) (cit. on pp. i, 9).
- [4] Ministero della Salute. *Fascicolo Sanitario Elettronico (FSE)*. 2025. URL: <https://www.fascicolosanitario.gov.it/> (visited on 08/11/2025) (cit. on pp. i, 9).
- [5] Governo Italiano. *PNRR e Fascicolo Sanitario Elettronico*. 2025. URL: <https://italiadomani.gov.it/it/home.html> (visited on 08/11/2025) (cit. on pp. i, 9).
- [6] Sanità Finanze. *FSE Portale riservato alle regioni: Abruzzo, Calabria, Campania, Sicilia e ai SASN*. 2025. URL: <https://fascicolosanitario.sanita.finanze.it/FseHomeWeb/accesso.xhtml> (visited on 08/11/2025) (cit. on pp. i, 9).
- [7] Dossier Médical Partagé. *Mon espace santé, l'espace numérique de santé des patients*. 2025. URL: <https://www.dmp.fr/> (visited on 08/11/2025) (cit. on pp. i, 9).
- [8] Sunjun Kweon et al. «Publicly shareable clinical large language model built on synthetic clinical notes». In: *arXiv preprint arXiv:2309.00237* (2023) (cit. on pp. i, 14, 17–19).
- [9] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. «Mimic-iv». In: *PhysioNet. Available online at: https://physionet.org/content/mimiciv/1.0/(accessed August 23, 2021)* (2020), pp. 49–55 (cit. on pp. i, 14, 32).

- [10] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. «Qlora: Efficient finetuning of quantized llms». In: *Advances in neural information processing systems* 36 (2023), pp. 10088–10115 (cit. on pp. i, 19).
- [11] Justin Xu et al. «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*. Association for Computational Linguistics, 2024, pp. 85–98. URL: <http://dx.doi.org/10.18653/v1/2024.bionlp-1.7> (cit. on pp. i, 20, 56, 58, 62, 78).
- [12] Jing Xu and PhysioNet. *Discharge Me: BioNLP ACL’24 Shared Task on Streamlining Discharge Documentation*. RRID:SCR_007345. 2024. DOI: 10.13026/0zf5-fx50. URL: <https://doi.org/10.13026/0zf5-fx50> (visited on 08/25/2024) (cit. on pp. i, 48, 58).
- [13] A. Sorita, P. M. Robelia, S. B. Kattel, C. P. McCoy, A. S. Keller, J. Almasri, M. H. Murad, J. S. Newman, and D. T. Kashiwagi. «The Ideal Hospital Discharge Summary: A Survey of U.S. Physicians». In: *Journal of Patient Safety* 17.7 (2021), e637–e644. DOI: 10.1097/PTS.0000000000000421 (cit. on pp. 2, 4).
- [14] Abstractive Health. *What is Discharge Summary—and Why is It Important?* <https://www.abstractivehealth.com/article/what-is-discharge-summary-and-why-is-it-important>. Accessed July 2025. 2023 (cit. on pp. 2, 3).
- [15] Agency for Healthcare Research and Quality. *Checklist for Discharge Summary Completion: Key Elements for Quality and Continuity of Care*. Tech. rep. Agency for Healthcare Research and Quality, 2007. URL: https://www.ahrq.gov/downloads/pub/advances2/vol2/advances-kind_31.pdf (cit. on pp. 2, 3).
- [16] Aaron M. Silver, Leigh Anne Goodman, Pooja Rangan, Romil Chadha, Michael Burton, Abey K. Thomas, and Christopher M. O’Donnell. «Bridge the Gap; Insights on Discharge Summaries from Inpatient Clinicians». In: *SHM Converge 2024 Abstracts*. Abstract 293; Journal of Hospital Medicine; July 27, 2025. 2025. URL: <https://shmabstracts.org/abstract/bridge-the-gap-insights-on-discharge-summaries-from-inpatient-clinicians> (cit. on p. 3).
- [17] Jordan Li, Tuck Yong, Paul Hakendorf, David Ben-Tovim, and Campbell Thompson. «Timeliness in Discharge Summary Dissemination is Associated with Patients’ Clinical Outcomes». In: *Journal of Evaluation in Clinical Practice* 19 (2011), pp. 141–148. DOI: 10.1111/j.1365-2753.2011.01772.x (cit. on p. 3).

- [18] Erik Hoyer, Charles Odonkor, Sumit Bhatia, Curtis Leung, Amy Deutschen-dorf, and Daniel Brotman. «Association between Days to Complete Inpa-tient Discharge Summaries with All-Payer Hospital Readmissions in Mary-land». In: *Journal of Hospital Medicine* 11.2 (2016), pp. 101–107. DOI: 10.1002/jhm.2556. URL: https://www.researchgate.net/publication/295910418_Association_between_days_to_complete_inpatient_discharge_summaries_with_all-payer_hospital_readmissions_in_Maryland (cit. on p. 3).
- [19] B. G. Arndt, J. W. Beasley, M. D. Watkinson, J. L. Temte, W. J. Tuan, C. A. Sinsky, and V. J. Gilchrist. «Tethered to the EHR: Primary Care Physician Workload Assessment Using EHR Event Log Data and Time-Motion Observations». In: *Annals of Family Medicine* 15.5 (2017), pp. 419–426. DOI: 10.1370/afm.2121 (cit. on p. 4).
- [20] Yuxuan Wu, Mingyue Wu, Changyu Wang, Jie Lin, Jialin Liu, and Siru Liu. «Evaluating the Prevalence of Burnout Among Health Care Professionals Related to Electronic Health Record Use: Systematic Review and Meta-Analysis». In: *JMIR Medical Informatics* 12 (2024), e54811. DOI: 10.2196/54811. URL: <https://medinform.jmir.org/2024/1/e54811> (cit. on pp. 4, 7).
- [21] Mariko M. Ching, Jennifer Lee, and Matthew Beecroft. «A Time-Motion Study of Emergency and Hospitalist Physicians in a Community Hospital Setting». In: *Journal of Wellness* 5.1 (2023), Article 7. DOI: 10.55504/2578-9333.1150. URL: <https://ir.library.louisville.edu/jwellness/vol5/iss1/7> (cit. on pp. 4, 6).
- [22] A. Momenipur and P. R. Pennathur. «Balancing Documentation and Direct Patient Care Activities: A Study of a Mature Electronic Health Record System». In: *International Journal of Industrial Ergonomics* 72 (2019), pp. 338–346. DOI: 10.1016/j.ergon.2019.06.012 (cit. on pp. 4, 6).
- [23] R. Tsopra et al. «Accuracy of Diagnoses in Discharge Summaries: A Cohort Study in Three Respiratory Wards». In: *Internal Medicine Journal* 48.4 (2018), pp. 460–466. DOI: 10.1111/imj.13908 (cit. on pp. 6, 7).
- [24] H. Cohen et al. «Electronic Versus Handwritten Discharge Summaries: Error and Omission Rates». In: *International Journal of Medical Informatics* 77.7 (2008), pp. 485–490. DOI: 10.1016/j.ijmedinf.2007.08.015 (cit. on pp. 6, 7).
- [25] A. Smith, B. Patel, and C. Lopez. «Completeness of Medication Information in Discharge Summaries: A Retrospective Audit». In: *BMC Health Services Research* 25 (2025), p. 112. DOI: 10.1186/s12913-025-12669-x (cit. on pp. 6, 7).

- [26] Nicole G. Weiskopf and Chunhua Weng. «Methods and Dimensions of Electronic Health Record Data Quality Assessment: Enabling Reuse for Quality Research». In: *Journal of the American Medical Informatics Association* 20.1 (2013), pp. 144–151. DOI: 10.1136/amiajnl-2011-000227 (cit. on p. 6).
- [27] Anonymous. *Challenges of Real-World Electronic Health Record Data: Fragmentation, Copy-Forward, and Missing Entries*. <https://arxiv.org/html/2507.14079v1>. Preprint; Accessed July 2025. 2025 (cit. on p. 6).
- [28] HIPAA Journal. *Why Do Criminals Target Medical Records?* Accessed: 2025-08-20. 2023. URL: <https://www.hipaajournal.com/why-do-criminals-target-medical-records/> (cit. on p. 7).
- [29] HIPAA Journal. *What Can Hackers Do with Medical Records?* Accessed: 2025-08-20. 2023. URL: <https://www.hipaajournal.com/what-can-hackers-do-with-medical-records/> (cit. on p. 7).
- [30] F-Secure. *Healthcare Data Breaches: Why Criminals Want Medical Records*. Accessed: 2025-08-20. 2024. URL: <https://blog.f-secure.com/health-care-data-breaches-why-criminals-want-medical-records/> (cit. on p. 7).
- [31] Associated Press. *Finnish Psychotherapy Center Files for Bankruptcy After Data Breach*. Accessed: 2025-08-20. 2021. URL: <https://apnews.com/article/europe-finland-health-data-breach-bankruptcy-0f05f7b4a7b84e39a2f5f1a7c46c3e2a> (cit. on p. 7).
- [32] BBC News. *Vastaamo Data Breach: Extortion of 30,000 Patients*. Accessed: 2025-08-20. 2021. URL: <https://www.bbc.com/news/technology-54692120> (cit. on p. 7).
- [33] Wired / Washington Post. *The Vastaamo Hack and Its Aftermath*. Accessed: 2025-08-20. 2022. URL: <https://www.wired.com/story/finland-mental-health-data-breach-vastaamo/> (cit. on p. 7).
- [34] Centers for Medicare & Medicaid Services. *The Health Insurance Portability and Accountability Act of 1996 (HIPAA)*. Online at <http://www.cms.hhs.gov/hipaa/> 1996 (cit. on p. 7).
- [35] European Parliament and Council of the European Union. *Regulation (EU) 2016/679 of the European Parliament and of the Council*. of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). May 4, 2016. URL: <https://data.europa.eu/eli/reg/2016/679/oj> (visited on 04/13/2023) (cit. on p. 7).

- [36] Nathalie A Smuha. «Regulation 2024/1689 of the Eur. Parl. & Council of June 13, 2024 (EU Artificial Intelligence Act)». In: *International Legal Materials* (2025), pp. 1–148 (cit. on p. 7).
- [37] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. «BLEU: a Method for Automatic Evaluation of Machine Translation». In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. ACL, 2002, pp. 311–318 (cit. on pp. 7, 20, 48, 49, 103).
- [38] Chin-Yew Lin. «ROUGE: A Package for Automatic Evaluation of Summaries». In: *Text Summarization Branches Out*. ACL, 2004, pp. 74–81 (cit. on pp. 7, 20, 48, 49, 103).
- [39] Satanjeev Banerjee and Alon Lavie. «METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments». In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT*. 2005, pp. 65–72 (cit. on pp. 7, 20, 48, 50, 103).
- [40] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. «BERTScore: Evaluating Text Generation with BERT». In: *International Conference on Learning Representations (ICLR)*. 2020 (cit. on pp. 7, 20, 48, 50, 51, 103).
- [41] Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. «AlignScore: Evaluating factual consistency with a unified alignment function». In: *arXiv preprint arXiv:2305.16739* (2023) (cit. on pp. 7, 20, 48, 51, 52, 103).
- [42] Wen-wai Yim, Yujuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. «Aci-bench: a novel ambient clinical intelligence dataset for benchmarking automatic visit note generation». In: *Scientific data* 10.1 (2023), p. 586 (cit. on pp. 7, 21, 48, 52, 103).
- [43] Jana Gundlack et al. «Patients’ Perceptions of Artificial Intelligence Acceptance, Challenges, and Use in Medical Care: Qualitative Study». In: *Journal of Medical Internet Research* 27 (2025), e70487 (cit. on p. 8).
- [44] Albert T Young, Dominic Amara, Abhishek Bhattacharya, and Maria L Wei. «Patient and general public attitudes towards clinical artificial intelligence: a mixed methods systematic review». In: *The lancet digital health* 3.9 (2021), e599–e611 (cit. on p. 8).
- [45] Zeinab Hamedani, Mohsen Moradi, Fatemeh Kalroozi, Ali Manafi Anari, Erfan Jalalifar, Arina Ansari, Behzad H Aski, Maryam Nezamzadeh, and Bardia Karim. «Evaluation of acceptance, attitude, and knowledge towards artificial intelligence and its application from the point of view of physicians and nurses: a provincial survey study in Iran: a cross-sectional descriptive-analytical study». In: *Health science reports* 6.9 (2023), e1543 (cit. on p. 8).

- [46] Sarah Negash, Jana Gundlack, Charlotte Buch, Timo Apfelbacher, Jan Schildmann, Thomas Frese, Jan Christoph, and Rafael T Mikolajczyk. «Physicians' Attitudes and Acceptance towards Artificial Intelligence in Medical Care: A Qualitative Study in Germany». In: *Frontiers in Digital Health* 7 (2025), p. 1616827 (cit. on p. 8).
- [47] Moustaq Karim Khan Rony et al. «Healthcare workers' knowledge and attitudes regarding artificial intelligence adoption in healthcare: A cross-sectional study». In: *Heliyon* 10.23 (2024) (cit. on p. 8).
- [48] European Commission. *European Health Data Space (EHDS)*. 2025. URL: https://health.ec.europa.eu/ehealth-digital-health-and-care/european-health-data-space_en (visited on 08/11/2025) (cit. on p. 9).
- [49] Stanford University. *Data Models - STARR (Stanford pRactice pAtient pRegistry)*. Accessed: 2025-09-04. Stanford University. URL: <https://starr.stanford.edu/data-models> (visited on 09/04/2025) (cit. on p. 9).
- [50] Stanford University. *Data Types - STARR (Stanford pRactice pAtient pRegistry)*. Accessed: 2025-09-04. Stanford University. URL: <https://starr.stanford.edu/data-types> (visited on 09/04/2025) (cit. on p. 9).
- [51] Cambridge University Hospitals. *eHospital - About Us*. Accessed: 2025-09-04. Cambridge University Hospitals. URL: <https://www.cuh.nhs.uk/about-us/our-structure/other-departments/ehospital/> (visited on 09/04/2025) (cit. on p. 9).
- [52] Ideal Health. *Milton Keynes University Hospital Case Study*. Accessed: 2025-09-04. Ideal Health. URL: <https://www.ideal-health.co.uk/case-studies/milton-keynes-university-hospital> (visited on 09/04/2025) (cit. on p. 9).
- [53] U.S. Department of Veterans Affairs. *VA names nine additional facilities that will deploy federal EHR in 2026*. Accessed: 2025-09-04. U.S. Department of Veterans Affairs. URL: <https://digital.va.gov/ehr-modernization/news-releases/va-names-nine-additional-facilities-that-will-deploy-federal-ehr-in-2026/> (visited on 09/04/2025) (cit. on p. 9).
- [54] Medical City Dallas. *Medical City Dallas's Tenant® Portal*. Accessed: 2025-09-04. Medical City Dallas. URL: <https://medicalcitydallas.com/main.cfm?pid=cssystem&sid=cservices> (visited on 09/04/2025) (cit. on p. 9).
- [55] MEDITECH. *Case Study: Princeton Community Hospital improves response time and physician efficiency with MEDITECH*. Accessed: 2025-09-04. MEDITECH. URL: <https://ehr.meditech.com/news/case-study-princeton-community-hospital-improves-response-time-and-physician-efficiency-with> (visited on 09/04/2025) (cit. on p. 9).

- [56] MEDITECH. *Scotland County Hospital Advances Community-Based Services with MEDITECH's Expanse EHR*. Accessed: 2025-09-04. MEDITECH. URL: <https://ehr.meditech.com/news/scotland-county-hospital-advances-community-based-services-with-meditech-s-expanse-ehr> (visited on 09/04/2025) (cit. on p. 9).
- [57] Vince Hartman. *Extractive vs. Abstractive Summarization in Healthcare*. Dec. 2022. URL: <https://www.abstractivehealth.com/article/extractive-vs-abstractive-summarization-in-healthcare> (visited on 08/12/2025) (cit. on p. 11).
- [58] Acalytica. *Breaking Down the Barriers to AI Adoption in Healthcare*. 2024. URL: <https://acalytica.com/blog/breaking-down-the-barriers-to-ai-adoption-in-healthcare> (visited on 08/13/2025) (cit. on p. 14).
- [59] Ctech. *Hospitals today face mounting challenges—staff shortages, financial pressure, and growing complexity. Yet much of the infrastructure used to manage operations is outdated or manual*. 2025. URL: <https://www.calcalistech.com/ctechnews/article/ry2ejr00bxe> (visited on 08/13/2025) (cit. on p. 14).
- [60] Eric Lehman et al. «Do we still need clinical language models?» In: *Conference on health, inference, and learning*. PMLR. 2023, pp. 578–597 (cit. on pp. 16, 17).
- [61] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. «BioGPT: generative pre-trained transformer for biomedical text generation and mining». In: *Briefings in Bioinformatics* 23.6 (Sept. 2022). ISSN: 1477-4054. DOI: 10.1093/bib/bbac409. URL: <http://dx.doi.org/10.1093/bib/bbac409> (cit. on p. 17).
- [62] Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. *BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model*. 2022. arXiv: 2204.03905 [cs.CL]. URL: <https://arxiv.org/abs/2204.03905> (cit. on pp. 17, 22).
- [63] Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. «Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine». In: *arXiv preprint arXiv:2308.09442* (2023) (cit. on p. 17).
- [64] Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. «Biomistral: A collection of open-source pretrained large language models for medical domains». In: *arXiv preprint arXiv:2402.10373* (2024) (cit. on p. 17).

- [65] Elliot Bolton et al. *BioMedLM: A 2.7B Parameter Language Model Trained On Biomedical Text*. 2024. arXiv: 2403.18421 [cs.CL]. URL: <https://arxiv.org/abs/2403.18421> (cit. on p. 17).
- [66] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. «Publicly available clinical BERT embeddings». In: *arXiv preprint arXiv:1904.03323* (2019) (cit. on p. 17).
- [67] Xi Yang et al. «Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records». In: *arXiv preprint arXiv:2203.03540* (2022) (cit. on pp. 17, 22).
- [68] Qiuhaio Lu, Dejing Dou, and Thien Nguyen. «ClinicalT5: A generative language model for clinical text». In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. 2022, pp. 5436–5443 (cit. on pp. 17, 22).
- [69] Bayan Altalla', Sameera Abdalla, Ahmad Altamimi, Layla Bitar, Amal Al Omari, Ramiz Kardan, and Iyad Sultan. «Evaluating GPT models for clinical note de-identification». In: *Scientific Reports* 15.1 (2025), p. 3852 (cit. on p. 18).
- [70] Maria Clara Saad Menezes et al. «The potential of Generative Pre-trained Transformer 4 (GPT-4) to analyse medical notes in three different languages: a retrospective model-evaluation study». In: *The Lancet Digital Health* 7.1 (2025), e35–e43 (cit. on p. 18).
- [71] Faezeh Ghaffari, Mostafa Langarizadeh, Ehsan Nabovati, and Mahdiah Sabery. «Effectiveness of ChatGPT for Clinical Scenario Generation: A Qualitative Study». In: *Archives of Academic Emergency Medicine* 13.1 (2025), e49 (cit. on p. 18).
- [72] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. «Capabilities of gpt-4 on medical challenge problems». In: *arXiv preprint arXiv:2303.13375* (2023) (cit. on p. 18).
- [73] Hanjae Kim, Hee Min Jin, Yoon Bin Jung, and Seng Chan You. «Patient-friendly discharge summaries in Korea based on ChatGPT: software development and validation». In: *Journal of Korean Medical Science* 39.16 (2024) (cit. on p. 18).
- [74] Josh Achiam et al. «Gpt-4 technical report». In: *arXiv preprint arXiv:2303.08774* (2023) (cit. on p. 18).

- [75] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. *Measuring Massive Multitask Language Understanding*. 2021. arXiv: 2009.03300 [cs.CY]. URL: <https://arxiv.org/abs/2009.03300> (cit. on p. 18).
- [76] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. «PubMedQA: A Dataset for Biomedical Research Question Answering». 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)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2567–2577. DOI: 10.18653/v1/D19-1259. URL: <https://aclanthology.org/D19-1259> (cit. on p. 18).
- [77] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. *MedMCQA : A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering*. 2022. arXiv: 2203.14371 [cs.CL]. URL: <https://arxiv.org/abs/2203.14371> (cit. on p. 18).
- [78] Zhengyun Zhao, Qiao Jin, Fangyuan Chen, Tuorui Peng, and Sheng Yu. «Pmc-patients: A large-scale dataset of patient summaries and relations for benchmarking retrieval-based clinical decision support systems». In: *arXiv preprint arXiv:2202.13876* (2022) (cit. on pp. 18, 19).
- [79] Khaled Saab et al. «Capabilities of gemini models in medicine». In: *arXiv preprint arXiv:2404.18416* (2024) (cit. on p. 18).
- [80] Lin Yang et al. «Advancing multimodal medical capabilities of Gemini». In: *arXiv preprint arXiv:2405.03162* (2024) (cit. on p. 18).
- [81] MIT Laboratory for Computational Physiology. *The PhysioNet Credentialed Health Data License, Version 1.5.0*. <https://physionet.org/content/mimiciv/view-license/0.4/>. This is the Data Use Agreement (DUA) governing the use of restricted PhysioNet datasets, such as MIMIC-IV. Copyright (c) 2025 MIT Laboratory for Computational Physiology. 2025 (cit. on pp. 18, 22, 33).
- [82] Guangyu Wang, Guoxing Yang, Zongxin Du, Longjun Fan, and Xiaohu Li. «ClinicalGPT: large language models finetuned with diverse medical data and comprehensive evaluation». In: *arXiv preprint arXiv:2306.09968* (2023) (cit. on p. 19).
- [83] Ye Chen, Igor Couto, Wei Cai, Cong Fu, and Bruno Dorneles. «SoftTiger: A Clinical Foundation Model for Healthcare Workflows». In: *arXiv preprint arXiv:2403.00868* (2024) (cit. on p. 19).

- [84] Ary L. Goldberger et al. «PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals». In: *Circulation [Online]* 101.23 (2000), e215–e220. DOI: 10.1161/01.CIR.101.23.e215. URL: <https://doi.org/10.1161/01.CIR.101.23.e215> (cit. on pp. 19, 33).
- [85] Augustin Toma, Patrick R Lawler, Jimmy Ba, Rahul G Krishnan, Barry B Rubin, and Bo Wang. «Clinical camel: An open expert-level medical language model with dialogue-based knowledge encoding». In: *arXiv preprint arXiv:2305.12031* (2023) (cit. on p. 19).
- [86] Hugo Touvron et al. «Llama 2: Open foundation and fine-tuned chat models». In: *arXiv preprint arXiv:2307.09288* (2023) (cit. on p. 19).
- [87] Karan Singhal et al. «Toward expert-level medical question answering with large language models». In: *Nature Medicine* 31.3 (2025), pp. 943–950 (cit. on p. 19).
- [88] Wei-Lin Chiang et al. «Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality». In: *See <https://vicuna.lmsys.org> (accessed 14 April 2023)* 2.3 (2023), p. 6 (cit. on p. 19).
- [89] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. «Alpaca: A strong, replicable instruction-following model». In: *Stanford Center for Research on Foundation Models*. <https://crfm.stanford.edu/2023/03/13/alpaca.html> 3.6 (2023), p. 7 (cit. on p. 19).
- [90] Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressemer. «MedAlpaca—an open-source collection of medical conversational AI models and training data». In: *arXiv preprint arXiv:2304.08247* (2023) (cit. on p. 19).
- [91] Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. «Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge». In: *Cureus* 15.6 (2023) (cit. on p. 19).
- [92] Dina Demner-Fushman, Sophia Ananiadou, Makoto Miwa, Kirk Roberts, and Jun’ichi Tsujii. «Proceedings of the 23rd Workshop on Biomedical Natural Language Processing». In: *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*. 2024 (cit. on p. 20).
- [93] Alistair Johnson, Luca Bulgarelli, Tom Pollard, Leo Anthony Celi, Roger Mark, and Steven Horng. *MIMIC-IV-ED (version 2.2)*. 2023. DOI: 10.13026/5ntk-km72. URL: <https://doi.org/10.13026/5ntk-km72> (cit. on p. 20).

- [94] Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. *MIMIC-IV-Note: Deidentified free-text clinical notes (version 2.2)*. 2023. DOI: 10.13026/1n74-ne17. URL: <https://doi.org/10.13026/1n74-ne17> (cit. on pp. 20, 35).
- [95] Luca Soldaini and Nazli Goharian. «Quickumls: a fast, unsupervised approach for medical concept extraction». In: *MedIR workshop, sigir*. 2016, pp. 1–4 (cit. on p. 21).
- [96] Hendrik Damm, Tabea MG Pakull, Bahadır Eryılmaz, Helmut Becker, Ahmad Idrissi-Yaghir, Henning Schäfer, Sergej Schultenkämper, and Christoph M Friedrich. «WisPerMed at "Discharge Me!": Advancing Text Generation in Healthcare with Large Language Models, Dynamic Expert Selection, and Priming Techniques on MIMIC-IV». In: *arXiv preprint arXiv:2405.11255* (2024) (cit. on pp. 21, 55, 62, 98).
- [97] Paul Landes, Kunal Patel, Sean S. Huang, Adam Webb, Barbara Di Eugenio, and Cornelia Caragea. «A New Public Corpus for Clinical Section Identification: MedSecId». In: *Proceedings of the 29th International Conference on Computational Linguistics*. Ed. by Nicoletta Calzolari et al. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 3709–3721. URL: <https://aclanthology.org/2022.coling-1.326/> (cit. on p. 21).
- [98] Paul Landes, Barbara Di Eugenio, and Cornelia Caragea. «DeepZensols: A deep learning natural language processing framework for experimentation and reproducibility». In: *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*. 2023, pp. 141–146 (cit. on p. 21).
- [99] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. «Lora: Low-rank adaptation of large language models.» In: *ICLR 1.2* (2022), p. 3 (cit. on p. 21).
- [100] Vimig Socrates, Thomas Huang, Xuguang Ai, Soraya Fereydooni, Qingyu Chen, R Andrew Taylor, and David Chartash. «Yale at “discharge me!”: Evaluating constrained generation of discharge summaries with unstructured and structured information». In: *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*. 2024, pp. 724–730 (cit. on p. 22).
- [101] Jinghui Liu, Aaron Nicolson, Jason Dowling, Bevan Koopman, and Anthony Nguyen. «e-Health CSIRO at "Discharge Me!" 2024: Generating Discharge Summary Sections with Fine-tuned Language Models». In: *arXiv preprint arXiv:2407.02723* (2024) (cit. on pp. 22, 55, 61).

- [102] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. «PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization». In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 5245–5263. DOI: 10.18653/v1/2022.acl-long.360. URL: <https://aclanthology.org/2022.acl-long.360/> (cit. on p. 22).
- [103] Mengxian Lyu, Cheng Peng, Daniel Paredes, Ziyi Chen, Aokun Chen, Jiang Bian, and Yonghui Wu. «UF-HOBI at " Discharge Me!": A Hybrid Solution for Discharge Summary Generation Through Prompt-based Tuning of GatorTronGPT Models». In: *arXiv preprint arXiv:2407.15359* (2024) (cit. on p. 22).
- [104] Cheng Peng, Xi Yang, Mengxian Lyu, Kaleb E Smith, Anthony Costa, Mona G Flores, Jiang Bian, and Yonghui Wu. «GatorTron and GatorTronGPT: large language models for clinical narratives». In: *AAAI 2024 Spring Symposium on Clinical Foundation Models*. 2024 (cit. on p. 22).
- [105] Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. «Dissecting contextual word embeddings: Architecture and representation». In: *arXiv preprint arXiv:1808.08949* (2018) (cit. on p. 50).
- [106] Ian Tenney, Dipanjan Das, and Ellie Pavlick. «BERT rediscovers the classical NLP pipeline». In: *arXiv preprint arXiv:1905.05950* (2019) (cit. on p. 50).
- [107] Aaron Grattafiori et al. «The llama 3 herd of models». In: *arXiv preprint arXiv:2407.21783* (2024) (cit. on p. 98).
- [108] Albert Q. Jiang et al. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL]. URL: <https://arxiv.org/abs/2310.06825> (cit. on p. 98).
- [109] U.S. Environmental Protection Agency. *Greenhouse Gases Equivalencies Calculator – Calculations and References*. Accessed: 2025-09-26. 2024. URL: <https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations> (cit. on p. 102).