POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Evaluating Large Language Models on Free-Text Data in an Italian Obstetric Context

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Author: **Gianluca Carta**

Student ID: 251385
Advisor: Prof. Maria Gabriella Signorini
Co-advisors: Giulio Steyde, Pierluigi Reali, Mark J. Carman
Academic Year: 2025-26

Abstract

In modern healthcare, it is common to have unstructured data in Electronic Health Records (EHRs). It often consists of free text characterized by abbreviations, acronyms, jargon, and inconsistent syntax. This lack of structure makes such data inaccessible for automated research. Due to the strict privacy regulations and hardware constraints of clinical environments, this thesis evaluates the performance of locally deployable open-source Large Language Models (LLMs) in correcting and extracting structured information from noisy, non-English medical text.

A modular, automated pipeline was developed to test seven model families (Gemma3, Llama3, MedGemma, Mistral, GPT-OSS, Qwen3, and DeepSeek) ranging from 4 to 70 billion parameters, executed locally via the Ollama framework. The models were evaluated on two main tasks: a correction task to fix typos and expand Italian obstetric acronyms, and an extraction task to retrieve 28 specific clinical fields, such as Apgar scores and blood gas values. Performance was measured against an expert-validated gold standard of 100 notes, analyzing various prompt engineering strategies (e.g., few-shot, positive instructions, domain-specific acronym lists, and prompt repetition) and using both string-based metrics and semantic embedding similarities.

The results demonstrated that local LLMs can successfully clean and extract clinical data, though capabilities strongly depend on architecture and prompt design. For note correction, directly injecting domain knowledge into the prompt—such as a list of common acronyms—significantly outperformed other techniques like few-shot prompting or simply scaling up the model size. For structured data extraction, some smaller models (notably Qwen3 Small) surprisingly matched or even outperformed their larger counterparts, proving that instruction-following capabilities often matter more than raw parameter counts. Among the tested architectures, Mistral and Gemma3 models provided the most favorable trade-off between high semantic accuracy and low inference latency, making them highly suitable for real-world deployment. Ultimately, this research establishes a concrete methodological foundation showing that small LLMs can successfully unlock the value of unstructured historical clinical data without the need for privacy-compromising cloud APIs or computationally expensive fine-tuning.

Keywords: Large Language Models (LLMs), Clinical Information Extraction, Unstructured Data, Prompt Engineering, Obstetric Prenatal Data

Sommario

Nella sanità moderna, la presenza di dati non strutturati all'interno delle cartelle cliniche elettroniche (EHR) è un fenomeno comune. Questi dati spesso consistono in testi liberi caratterizzati da abbreviazioni, acronimi, gergo tecnico e una sintassi incoerente. La mancanza di struttura rende queste informazioni inutilizzabili per la ricerca. A causa delle rigide normative sulla privacy e dei limiti hardware degli ambienti clinici, questa tesi valuta le prestazioni di Large Language Models (LLM) open-source, eseguibili localmente, nella correzione ed estrazione di informazioni strutturate da testi medici rumorosi in lingua italiana.

Abbiamo sviluppato una pipeline modulare e automatizzata per testare sette famiglie di modelli (Gemma3, Llama3, MedGemma, Mistral, GPT-OSS, Qwen3 e DeepSeek) con un numero di parametri compreso tra 4 e 70 miliardi, eseguiti localmente tramite il framework Ollama. I modelli sono stati valutati su due task principali: uno di correzione per eliminare refusi ed espandere acronimi ostetrici, e uno di estrazione per estrarre 28 valori clinici, come i punteggi Apgar e i valori dell'emogasanalisi. Le prestazioni sono state misurate rispetto a un gold standard di 100 note validate da esperti, analizzando diverse strategie di prompt engineering (ad esempio, few-shot, istruzioni positive, liste di acronimi dominio-specifici e ripetizione del prompt) e utilizzando sia metriche basate sulle stringhe che metriche basate sulla semantica (similarità di embeddings).

I risultati dimostrano che gli LLM locali sono in grado di pulire ed estrarre con successo i dati clinici, sebbene le capacità dipendano fortemente dall'architettura e dalla progettazione del prompt. Per la correzione delle note, l'inserimento diretto di conoscenza di dominio nel prompt — come un elenco di acronimi comuni — ha superato altre tecniche come il few-shot prompting o il semplice aumento delle dimensioni del modello. Per l'estrazione di dati strutturati, alcuni modelli più piccoli (in particolare Qwen3 Small) hanno eguagliato o superato le loro controparti più grandi, dimostrando che le capacità di seguire le istruzioni (instruction-following) possono influire più del solo numero di parametri. Tra le architetture testate, i modelli Mistral e Gemma3 si sono dimostrate il miglior compromesso tra alta accuratezza semantica e bassa latenza di inferenza, rendendoli particolarmente adatti per l'implementazione in contesti reali. In definitiva, questa

ricerca stabilisce una solida base metodologica dimostrando che gli LLM di piccole dimensioni possono valorizzare con successo i dati clinici storici non strutturati, senza ricorrere ad API cloud che potrebbero compromettere la privacy o a processi di fine-tuning computazionalmente onerosi.

Parole chiave: Large Language Models (LLM), Estrazione di Informazioni Cliniche, Dati Non Strutturati, Prompt Engineering, Dati Ostetrici Prenatali

Contents

1 | Introduction

1.1. Context: NLP and Clinical Research

In the modern clinical landscape, the digital transformation of healthcare has led to a paradoxical situation: a massive presence of Electronic Health Records (EHRs) but a profound lack of utilizable data for research [1]. While databases are designed to store structured fields like a patient's age or predefined laboratory values, a lot of data is stored in free text formats [1, 2] which are often a collection of facts considered important to remember by physicians. They are often recorded as shorthand fragments rather than complete sentences, containing abbreviations, medical jargon, grammatical errors, and non-standard formatting [1, 2]. This creates textual data that is "invisible" to automated analysis: information that is technically present in the system but not readily accessible to statistical tools [1].

Beyond the technical difficulty of reading text, a systemic lack is due to the fragmented nature of the clinical landscape. Hospitals and clinics tend to keep their datasets private and isolated. Strict privacy regulations mean there is often no data integration between different institutions [3–5]. However, even if privacy were not a concern, a massive standardization barrier remains. Global data integration typically requires mapping information to a common language (usually English) or to formal Medical Ontologies (such as SNOMED CT or ICD-10) [6, 7]. For these types of mapping to function, they require input text that is disambiguated and standardized: features that noisy clinical shorthands entirely lack. Consequently, there is a new need for a preliminary "text cleaning and reconstruction" phase, where preprocessing techniques can improve downstream clinical concept extraction without information loss [2].

The unstructured nature of medical notes and fragmented nature of medical datasets together create a combination of problems for which Large Language Models (LLMs) emerge as a logical solution. Unlike traditional rule-based systems that struggle with the features of human writing [8], LLMs are inherently able to interpret, understand, and reconstruct text by leveraging the semantic patterns learned during their training.

However, most contemporary research on LLMs focuses on cloud-based models (like GPT-5) and clean, long-form English text [9]. There is comparatively little focus on small-scale models and short, noisy non-English text, particularly in resource-constrained clinical environments [10].

To address this gap, we investigate a concrete and representative scenario: Italian-language obstetric notes from a major hospital, which exemplify the precise combination of small-scale model deployment, noisy text, and non-English context.

1.2. Scenario and Problem Statement

The challenges of unstructured and fragmented data described in the previous section find a concrete and representative manifestation in the field of obstetrics. Research in this domain frequently faces a chronic scarcity of utilizable data, particularly concerning the prenatal phase. One of the most significant resources currently available is the dataset provided by the Federico II Hospital in Naples, which has been the subject of prior structured data curation efforts [11]. This dataset includes thousands of prenatal visit reports. Each record contains structured physiological data, such as signals like Fetal Heart Rate (FHR) and other biometric data, and a "Note" field (Nota).

Such notes were not intended to be formal documents: they were conceived as quick memos for the physician in view of the patient's next visit. Consequently, only the most critical information and values were recorded. This utility-driven approach resulted in several linguistic challenges that limit their direct use for research or downstream computational applications:

- **Acronyms and Abbreviations of Jargon:** Abbreviations and acronyms of words from obstetric Italian jargon, recognizable only by people who know the context and deal with certain concepts weekly or daily. Examples include:
  - `"iii gr"` for `"terza gravidanza"` (third pregnancy)
  - `"ivg"` for `"interruzione volontaria di gravidanza"` (voluntary termination of pregnancy)
  - `"tc"` for `"taglio cesareo"` (cesarean section)
- **Syntactic Noise:** A near-total absence of formal punctuation. It is common to find separate concepts without a comma in between, or even distinct words merged without a space. Examples include:
  - `"ii gr n peso 3300 sesso f c gen b"` for `"terza gravidanza, neonato:`

`peso 3300 grammi, sesso femminile, condizioni generali buone"` (second pregnancy, newborn: weight 3300 grams, female, good general conditions)

– `"nfpeso3300a78"` for `"neonato: sesso femminile, peso 3300 grammi, apgar 7-8"` (newborn: female, weight 3300 grams, apgar 7-8)

Any enhancement in the quality of these notes could help both the medical NLP field and the obstetric research. For example, it could facilitate the creation of Italian ontologies and the mapping of local data to international English-based standards [12]. Moreover, recent work has demonstrated that systematic standardization of obstetric diagnostic terminology using LLMs with prompt engineering can significantly improve data quality and downstream analysis [13, 14].

This thesis addresses the challenge of clinical data quality by investigating whether locally-deployable LLMs can effectively clean, correct, and extract structured information from such noisy obstetric notes - all of this while respecting the privacy and computational constraints of real clinical environments.

1.3. Methodology

Considering the premises established in the previous sections, our work positions itself at the intersection of clinical Natural Language Processing and resource-constrained language modeling. The proposed framework is designed to be modular, reproducible, and easily adaptable to diverse clinical environments. The goal is to produce empirical results that contribute to both the obstetric domain and the broader field of medical AI.

The research follows an experimental methodology, structured into a workflow composed of the following stages:

1. **Data Understanding and Exploratory Analysis:** We performed exploratory data analysis of the dataset provided by the Federico II Hospital. We investigated key properties such as note length distributions and most frequent words. To fully understand the meaning of the physician notes, we were helped by a professional obstetrician.

2. **Definition of Tasks:** To evaluate the models' ability to interpret high-noise clinical shorthand, we defined two primary objectives:

   - **Correction:** This task assesses the model's ability to identify typographical errors and expand context-specific acronyms and abbreviations into their full, formal Italian equivalents.

- **Extraction:** This task focuses on Information Extraction (IE). We identified a set of "interesting fields" - e.g., laboratory results and neonatal health indicators - to evaluate whether the models could successfully recognize and isolate these values from the unstructured text.

3. **Evaluation Metrics and Ground Truth:** Under the supervision of an obstetrician, we developed a gold-standard test set of 100 notes. For each note, the expert provided a corrected version and the expected structured values. Correction metrics employed word-level binary metrics (like Accuracy and F1-score), sequence similarity (BLEU), and semantic embedding similarity to measure how closely the model's output matched the expert's interpretation. Extraction metrics were based on the model's precision in identifying the presence of specific values and its ability to format them into a structured, machine-readable output.

4. **Selection of Large Language Models:** To respect the privacy and infrastructural constraints of a hospital setting, we focused exclusively on Open-Source, locally-deployable LLMs. We selected models ranging from 4 to 40 GB in size to simulate realistic hardware scenarios [10]. Using the Ollama framework [15, 16], we experimented with models from seven distinct families: Gemma3, Llama3, MedGemma, Mistral, GPT-OSS, Qwen3, and DeepSeek. For each family, we selected up to three models different in size ("small", "medium", "large").

5. **Prompt Engineering and Design:** Recognizing the linguistic specificities of the data, we adopted Italian-language prompting to minimize the "translation gap" during processing. For each task we started from a Basic prompt, made of a foundational set of instructions providing context and a description of the task. We enriched it to make new prompts where we implemented Prompt Engineering techniques (e.g., Few-Shot, Positive Instructions) to isolate which strategies most significantly impact performance in a medical context.

6. **Pipeline Implementation:** We developed a modular and extensible software pipeline designed to be executed via simple terminal commands, with an Execution Module that automates the iteration over the list of models and prompts, and an Evaluation Module that automatically computes scores for every model-prompt combination on the pre-determined metrics.

7. **Execution and Multi-Dimensional Analysis:** Following the experimental runs, we conducted a rigorous statistical analysis of the resulting scores. Performance was analyzed along three primary axes: (i) Model Family, (ii) Model Size, and (iii) Prompt Variation.

8. **Synthesis and Conclusions:** In the final stage, we synthesized the quantitative scores to discuss the implications for medical NLP, identifying which configurations offer the highest reliability for clinicians.

1.4. Contributions

This work positions itself at a critical gap in medical NLP literature: the evaluation of small LLMs on non-English, highly noisy medical text under resource constraints. The specific contributions are:

1. **A comprehensive experimental methodology:** We design a reusable, modular pipeline for medical text processing that combines correction, extraction and evaluation. This methodology is generalizable to other medical contexts, languages, and data processing tasks.
2. **An evaluation of small LLMs on Italian obstetric notes:** We provide a systematic study comparing multiple small LLM families (ranging from 4B to 70B parameters) on real, unstructured Italian medical text. This fills a significant gap, as most LLM research focuses on large models and clean English text.
3. **Multi-dimensional analysis of design factors:** By systematically varying model family, model size, and prompt strategies, we provide evidence-based recommendations for deploying small LLMs in resource-constrained clinical environments and insights into which model families perform best, how prompt engineering impacts performance, and what performance levels can realistically be achieved.

1.5. Related Work

1.5.1. Fetal Monitoring and Cardiotocography

Fetal heart rate (FHR) and uterine contractions (TOCO) recording via cardiotocography (CTG) is a standard method for monitoring fetal status during gestations. The signals recorded are used to detect fetal distress (or acidemia) [17] and Intrauterine Growth Restriction (IUGR) [18–20]. While traditional analysis relied on visual interpretation, recent shifts toward computerized quantitative analysis have improved the discrimination between healthy and pathological fetuses [11, 18, 21].

The dataset used in this thesis, sourced from the Federico II Hospital in Naples, has been previously utilized for signal-based analysis [11, 21, 22]. Prior work focused on deep

learning approaches to estimate gestational age and detect developmental deviations from CTG recordings [11, 20]. These studies used the "Note" field primarily for label generation. Our work shifts the focus from the signals to the textual records, treating them as a rich information source of data that can be used as the direct input for future analysis.

1.5.2. Clinical NLP: from Rules to LLMs

Historically, extracting structured data from clinical text relied on labor-intensive rule-based systems or manual annotation [23, 24]. The transition to machine learning, specifically Named Entity Recognition (NER) using models like BERT, improved scalability but often struggled with the "noisy" nature of real-world documentation [2, 23].

The emergence of LLMs has transformed clinical NLP by leveraging semantic patterns to reconstruct noisy text. Recent evidence suggests that LLM-based preprocessing - correcting spelling, expanding acronyms, and standardizing terminology - significantly enhances downstream concept extraction without information loss [2].

1.5.3. LLMs in Obstetrics and Gynecology

Recent research highlights the transformative potential of LLMs in obstetrics, particularly in extracting "hidden" predictors from narrative text and standardizing clinical terminology.

A study by Wang et al. (2024) [13] evaluated the use of LLMs (such as ChatGLM2 and Qwen-14B) for mapping obstetric diagnoses to the ICD-10 observation domain using data collected from the People's Hospital of Guangxi Zhuang Autonomous Region. This research shares a core objective with this thesis: the standardization of unstructured obstetric data to improve research value. However, while Wang et al. focus on the high-level task of mapping terms and notes to global codes, this thesis addresses the preliminary and more granular challenge of text correction and acronym expansion specifically within the Italian linguistic context.

The clinical utility of such extraction methods is further demonstrated by Woo et al. (2025) [14], who applied LLMs to prenatal notes from the St. Luke's University Health Network (Pennsylvania and New Jersey) to predict postpartum hemorrhage (PPH). Their "LLM-extract" pipeline is methodologically in common with the extraction task of this thesis: both use LLMs to transform "invisible" narrative features into structured, computable data. The key difference lies in the ultimate goal: the PPH study is predictive (real-time risk stratification), while this thesis only focuses on the modular automation of

the cleaning process. On the other side, unlike the PPH study which uses English-language health data, this thesis addresses the additional complexity of the Italian language.

Furthermore, the PERFORM study (2025) [25] established a benchmark for clinical reasoning, showing that LLMs can surpass resident physicians in diagnostic accuracy with minimal performance loss between English and Italian. However, a critical gap remains: the PERFORM study utilized standardized, examination-style scenarios rather than authentic clinical records. This thesis moves beyond such "perfect" data to address the linguistic noise of real-world notes.

1.6. Structure of Thesis

- **Chapter 2: Goals and Requirements** describes the objectives of the study, defining requirements for the proposed system in a clinical environment.
- **Chapter 3: Data Exploration** provides an analytical overview of the Federico II Hospital dataset.
- **Chapter 4: Approach and Implementation** explains the design decisions, including the annotation of the gold standard, the construction of the LLM inference pipeline, and the variety of prompt engineering strategies tested.
- **Chapter 5: Evaluation and Results** presents a multi-dimensional analysis of model performance (family, size, and prompt), detailing the results of the correction and extraction tasks.
- **Chapter 6: Conclusion and Future Work** synthesizes the findings, discusses the broader implications for medical NLP, and identifies future research opportunities.

2 | Goals and Requirements

2.1. Refined Problem Statement

Building upon the problem statement introduced in Section 1.2, we now refine the core problems we address:

- **Data Quality Problem:** Obstetric notes in the Federico II Hospital dataset contain substantial noise, domain-specific abbreviations, and inconsistent formatting that makes them unsuitable for automated analysis or integration with medical research databases.
- **Methodological Problem:** Many recent approaches to medical text processing either assume high-quality data or rely on cloud-based LLMs, neither of which is feasible for privacy-sensitive clinical environments working with non-English, domain-specific text.
- **Literature Gap:** There is a lack of empirical evaluation of small, open-source LLMs on noisy, non-English medical text. Most contemporary LLM research focuses on cloud-based models and/or clean text.

The specific focus of this thesis is to investigate whether locally-deployable LLMs can effectively support two key tasks:

1. **Correction Task:** Identifying and correcting typographical errors, expanding abbreviations and acronyms, and normalizing text formatting to produce more standardized medical notes.
2. **Extraction Task:** Identifying and extracting predefined medical information fields (e.g. laboratory values, gestational pathologies) from unstructured note text.

2.2. Goals

The ultimate goals of this thesis are:

1. **To establish a benchmark:** Provide an empirical evaluation of how locally deployable LLMs (4 to 70 billions parameters) perform on Italian obstetric text correction and information extraction tasks, establishing baseline performance metrics for future work.
2. **To identify best practices:** Through systematic experimentation, determine which combinations of model family, model size, and prompt engineering techniques yield the best performance while respecting computational constraints.
3. **To create a reusable methodology:** Develop a modular pipeline and evaluation framework that can be adapted to other medical contexts, languages, and clinical settings.
4. **To process large-scale clinical data:** Apply the identified best practices to clean and extract structured information from one of the world's largest cardiotocography (CTG) databases.

2.3. Requirements

Based on the goals above, we identify the following concrete requirements:

- **Privacy:** All models must be open-source and locally deployable, with no reliance on cloud services or external APIs.
- **Computational Efficiency:** Selected models must be executable on standard consumer hardware (8-40 GB RAM/VRAM).
- **Response Time:** Model inference time cannot exceed a few seconds per note to be practically useful for clinical workflows.
- **Modularity:** The pipeline must be modular and extensible to support new models, prompts, and metrics without major refactoring.

3 | Data Exploration

As previously mentioned, the original dataset originates from Federico II Hospital in Naples. It was constructed between 2013 and 2021 [11], and then continuously updated. For this thesis, most of the entries of the dataset with a "note" field (26,934 entries) have been used.

The version of the dataset utilized in this study was already anonymized and consisted of structured fields (e.g., gestational age, maternal age), four signal fields (FHR, TOCO, FMP, QUALITY), and an unstructured free-text field containing obstetric notes. The notes were obtained through aggregation: when a patient had multiple visits, the corresponding notes were concatenated into a single text field.

Consequently, while the dataset contained almost 27,000 entries, the number of unique notes was considerably smaller:

- 26,934 notes
- 7,886 unique notes

To obtain a fundamental yet comprehensive understanding of the data, we conducted a preliminary analysis using both classical Natural Language Processing techniques (length distributions, word frequencies, and word clouds) and a qualitative approach (expert review).

3.1. Classical NLP Exploration

Note Length Distributions

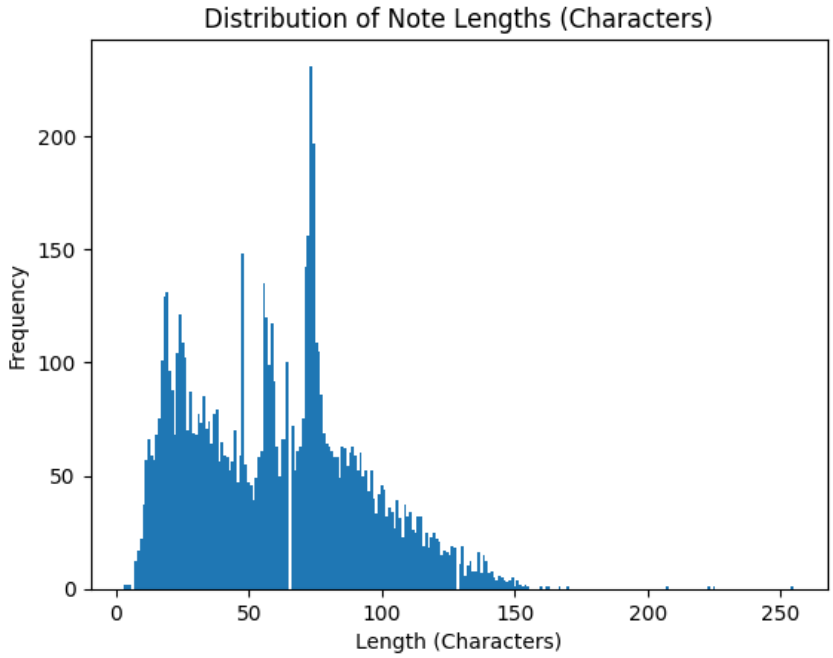


Figure 3.1: Length distribution of notes in the dataset, in number of characters.

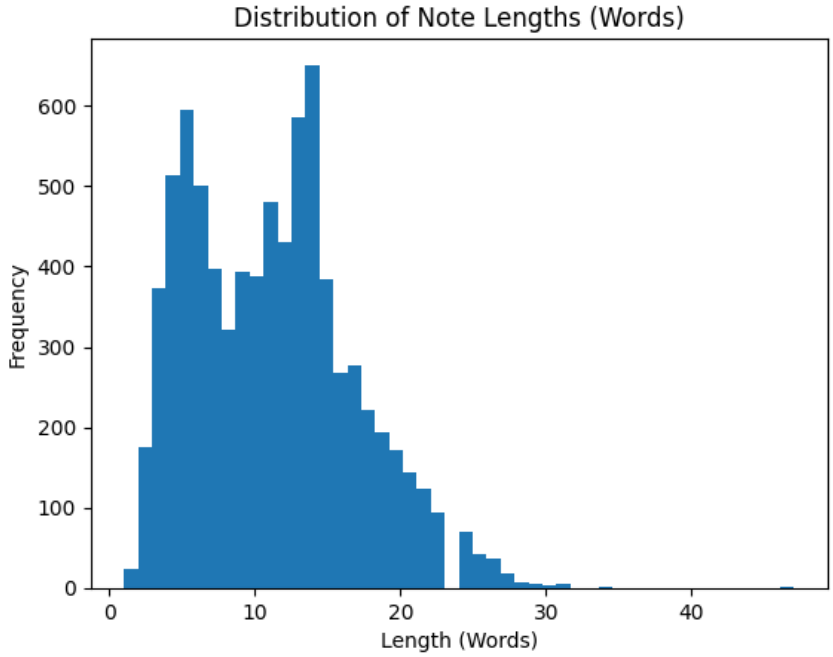


Figure 3.2: Length distribution of notes in the dataset, in number of words.

As it is possible to notice from the plots of the length distribution, most of the notes are short: only 9% of them has at least 20 words. And only 10 notes have at least 30 (0.13% of the notes). On the other side, having a big number of notes in the dataset, we can still count on a good quantity of medium-long notes: for example, the notes with 18 or more words are more than 1,000.

Word Length Distribution

Considering the "original" vocabulary (10,016 unique words) the length distribution of the words - measured in number of character - is shown in Fig. 3.3:

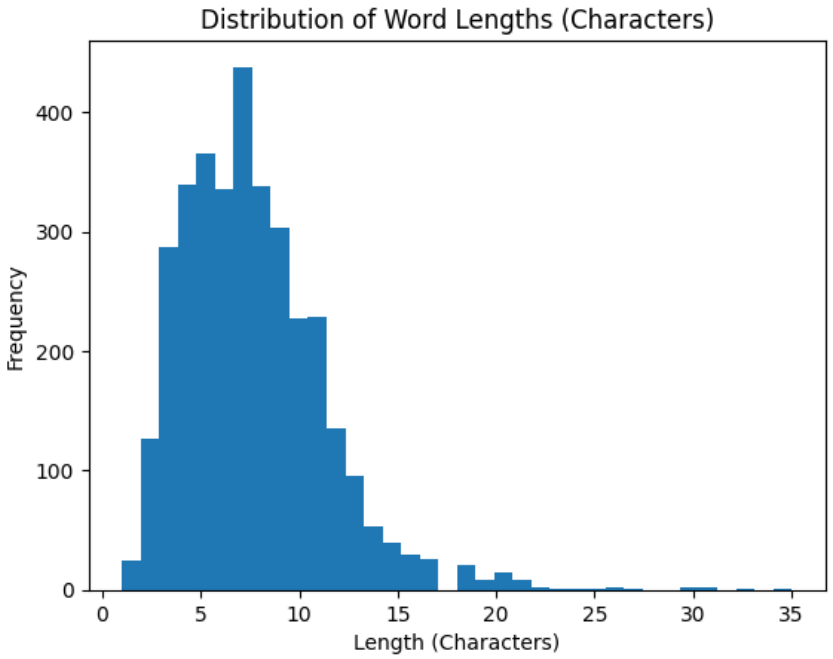


Figure 3.3: Length distribution of the words in the dataset's notes, in number of characters.

The plot reveals that many words are short (up to 5 characters), which is likely due to the prevalence of abbreviations in the medical notes. On the other side, there are also outliers with large lengths (up to 35 characters), which result from inadequate handling of punctuation and word boundaries.

For instance, the longest word in the dictionary is "variabiletiroiditehcvpiastrinopenia", which exemplifies a set of *words* that are actually multiple words concatenated without spaces, as we will discuss in the Qualitative Exploration section.

Vocabulary Size

Considering only unique words (case-insensitive), our vocabulary counts 10,016 words. Examining a few examples reveals how punctuation and numbers affect vocabulary size:

- "example" and "example)" are treated as two distinct words
- Numbers are sometimes attached to words (e.g., peso3000), creating additional vocabulary entries

It is possible to find that 65.2% of words contain at least one punctuation character. Additionally, 60.03% of words contain at least one digit. Their intersection represents 49.78% of the vocabulary.

For investigation purposes, we decided to remove any punctuation symbol from the words and count again: the vocabulary size decreased to 8,366 words, revealing that 16.47% of the vocabulary consists of duplicates with different punctuation (e.g., "p.s." and "ps").

Removing numbers (words containing only digits) further reduced the vocabulary to 4,857 entries. And removing all digits from the remaining words yielded a final vocabulary size of 3,428 words. For example, "peso3000" and "peso" were merged into the single word "peso".

Frequent Words

In the following page we are reporting the 20 most frequent words and their frequencies in three different cases:

- considering the original notes
- considering the notes without punctuation
- considering the notes without punctuation and without digits

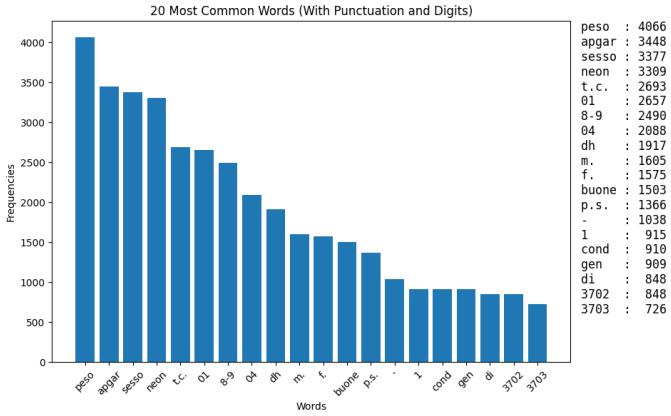


Figure 3.4: The 20 most frequent words in the original vocabulary, and their respective frequency.

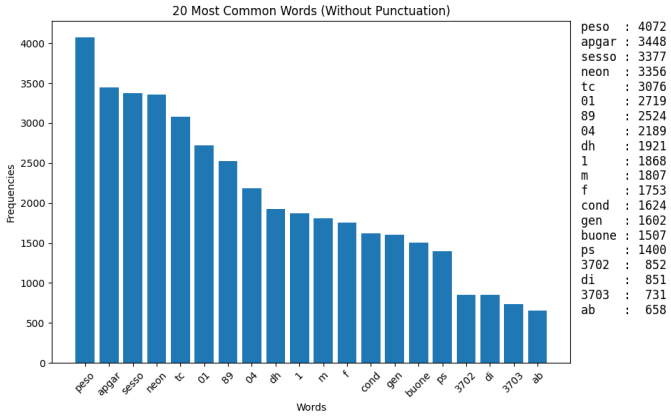


Figure 3.5: The 20 most frequent words after removing punctuation from the original vocabulary, and their respective frequency.

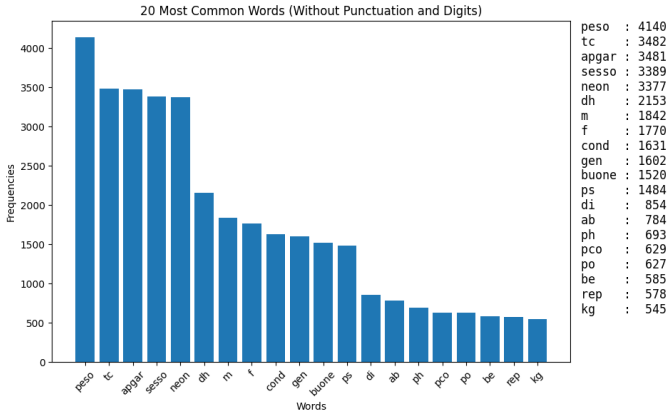


Figure 3.6: The 20 most frequent words after removing punctuation and digits from the original vocabulary, and their respective frequency.

As it's possible to notice, the actual words - like peso and apgar - kept gaining counts, showing how frequently they were paired with punctuation or digits.

An interesting case is the abbreviation for Taglio Cesareo (t.c.), whose frequency increased from 2,693 to 3,076 following punctuation removal, and finally to 3,482 after digits were also removed. The initial increase comes from merging "t.c." with "tc". The second gain, however, is due to instances where "tc" was adjacent to digits without spacing (e.g., "t.c.12/12/12", which becomes "tc121212" after punctuation removal and simply "tc" once digits are excluded too).

Also, it is nice to notice groups of related words having consistent frequencies:

- **cond gen buone**: This common abbreviation for *condizioni generali buone* (Italian for "good general conditions") shows nearly identical frequencies across all three terms.
- **sesso**: The frequency of the term *sesso* (sex) closely approximates the combined frequencies of *m* and *f* (representing *maschile* and *femminile* - male and female, respectively).

To provide an intuitive visualization of the most frequent terms, we generated word clouds for the three vocabulary variants. The frequency growth of "tc" is clearly visible, as is a similar pattern for "8-9", representing typical Apgar scores that frequently appear as "8-9", but sometimes are written as "8/9" or even "89".

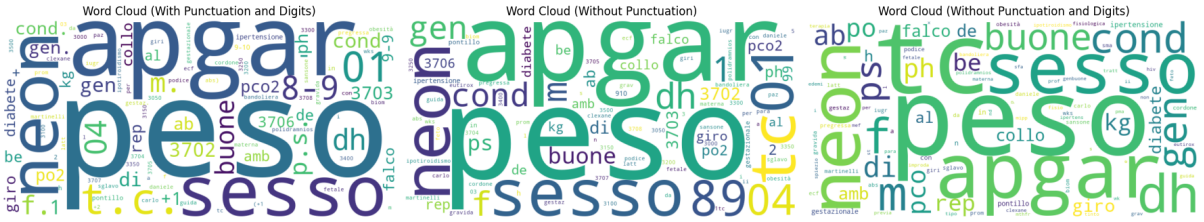


Figure 3.7: Word Clouds: built with the original vocabulary (left), after removing punctuation (center), and after removing also digits (right).

A note about stopwords

We have chosen not to remove stop words, as they may be present but as abbreviations of other actual words. Furthermore, they do not appear among the most frequent words in our dataset.

3.2. Qualitative Exploration

With the assistance of a professional obstetrician and a researcher working on fetal monitoring, we conducted a sample-based visual analysis to identify the most frequent abbreviations and clinically relevant values in the notes. The former, drawing on her experience as an obstetrician, provided a comprehensive list of the most common acronyms in obstetric practice, and in particular in these notes, along with their definitions. The latter, leveraging his expertise with the dataset and obstetric research, assisted in identifying the clinical values that frequently appear in the notes.

These two resources can be found in the Appendix (A.1, A.2) and will be instrumental in designing the prompts in the implementation phase (4.2.2, 4.2.3).

Obstetric Acronyms

Below are representative examples of the cited acronyms:

- **p.s.** for parto spontaneo (spontaneous delivery)
- **t.c.** for taglio cesareo (cesarean section)
- **ivg** for interruzione volontaria di gravidanza (voluntary termination of pregnancy)
- **ab/abs** for aborto/aborto spontaneo (abortion/spontaneous abortion)

Relevant Values

The most clinically relevant values identified can be grouped in the following categories:

Administrative Data: Administrative data related to the patient and the hospital visit, such as clinical codes, ward information, and physician details.

Neonatal Measurements: Physical and health measurements of the newborn, including weight, length, head circumference, Apgar scores [26], and gender.

Neonatal Blood Gas and Metabolic Values: Laboratory values related to blood chemistry, including pH levels, gas pressures (CO_2 and O_2), bicarbonate concentrations, base excess measurements, and lactate levels.

Maternal Health: Health metrics and conditions of the mother, including maternal weight, weight gain during pregnancy, and presence or absence of conditions such as hypertension, diabetes, and obesity.

4 | Approach and Implementation

Aligned with the goals described in Chapter 2: Goals and Requirements, we tasked several Large Language Models with correcting and extracting specific values from obstetric notes. Using an expert-validated "ground truth" as a reference, we designed and tested a variety of prompts for two specific tasks:

- **Correction:** given a note, the model identifies errors, abbreviations and acronyms, then corrects or expand them into a correct version of the input note
- **Extraction:** given a note, the model identifies and extract a list of pre-determined values (the ones discussed in the previous chapter's section "Relevant Values" 3.2)

Each model–prompt combination was evaluated using predefined metrics, and the results were analyzed along multiple dimensions: model family, model size, and prompt design. In this chapter we will discuss the automatization of the several phases, while the cited metrics and the multi-dimensional analysis will be discussed in Chapter 5: Evaluation and Results.

4.1. Design Decisions

4.1.1. Expert-Driven Ground Truth Construction

Given the low quality and high specificity of the obstetric notes, we opted against synthetic data in favor of expert-led validation. We worked with a practicing obstetrician to manually build a test set, comprising both corrected notes and structured data fields. This approach ensured that our evaluation of the models would be based on genuine clinical knowledge.

To streamline the annotation process, we selected 100 notes from the dataset and designed clear annotation interfaces to guide the physician through two distinct tasks. The note selection strategy combined the 20 longest notes in the dataset with 80 randomly sampled ones. This choice guaranteed a satisfying amount of clinical information available for evaluation while also preserving variability in note length, reflecting the original length

distribution.

The two Graphical User Interfaces we designed are:

- **Correction GUI:** The physician reads the original noisy note and writes a clinical-grade corrected version
- **Extraction GUI:** The physician identifies and populates structured fields relevant to the obstetric domain (the list of relevant fields discussed in 4.2.3)

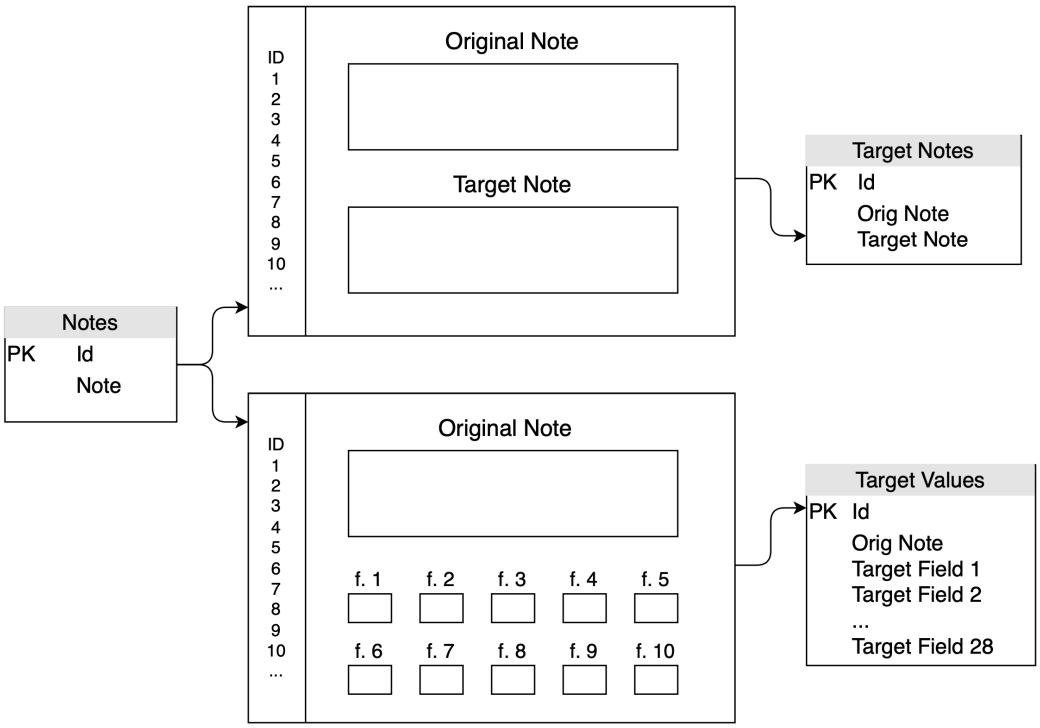


Figure 4.1: The conceptual design of the annotation interfaces used for test set creation: Correction GUI (top) and Extraction GUI (bottom), with input and respective outputs.

4.1.2. Automated Experimentation

To systematically evaluate how various models and prompting strategies perform, we conducted a multi-dimensional analysis across model families, model sizes, and prompt designs. Manual evaluation of all possible combinations was computationally and operationally infeasible. We therefore automated both the inference pipeline and the evaluation process:

- **Inference Automation:** For each task, a unified pipeline loads each considered model and systematically applies all prompt variations to the entire test set, capturing outputs and execution metadata

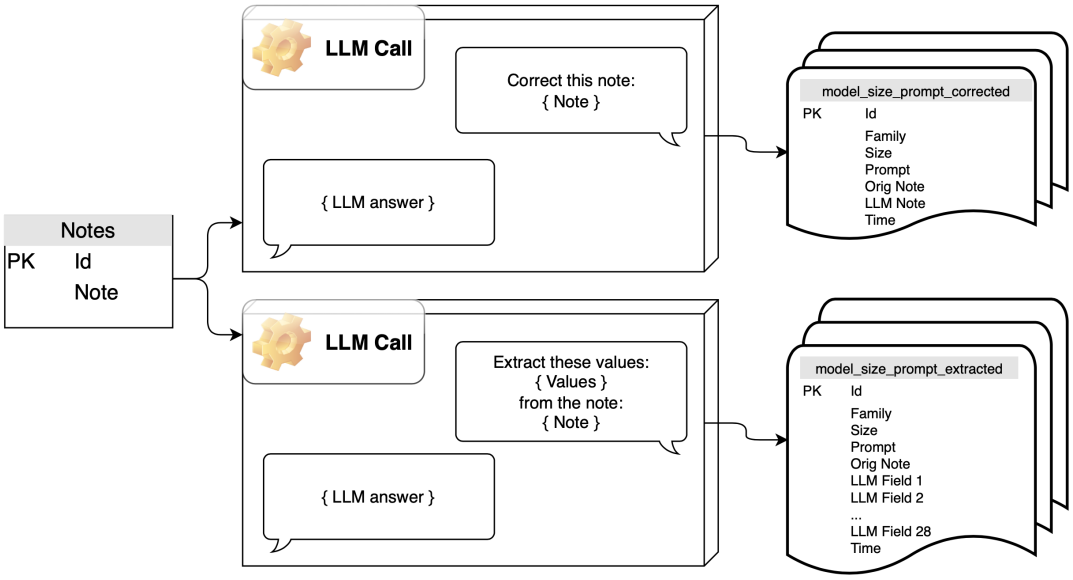


Figure 4.2: LLMs calls pipelines, with inputs and outputs, for Correction (top) and Extraction (Bottom)

- **Evaluation Automation:** A separate evaluation system computes all metrics against the gold standard, storing results in structured files for analysis.
  To ensure the statistical robustness of these findings, the calculated scores are processed via bootstrapping, with the resulting mean of means being recorded as the final representative value.

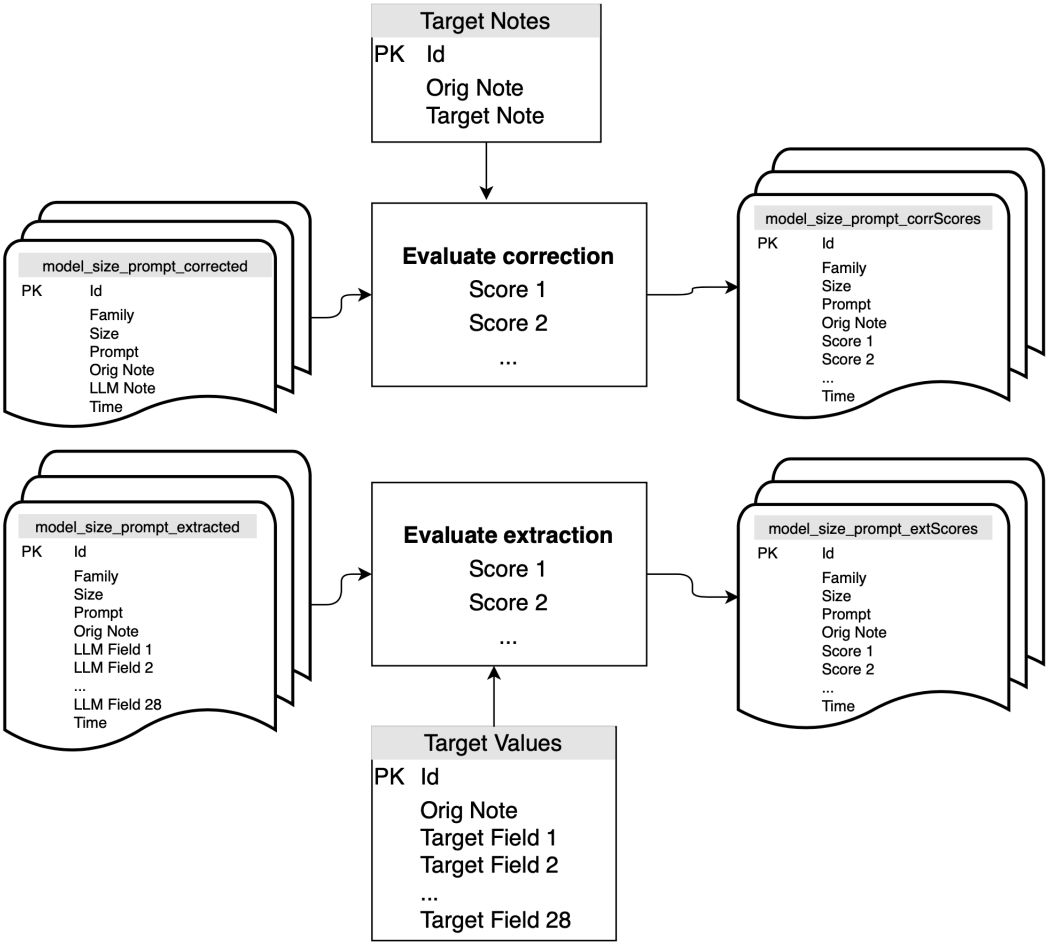


Figure 4.3: Evaluation pipelines, with inputs and outputs, for Correction (top) and Extraction (Bottom)

4.1.3. Multi-Dimensional Performance Analysis

The final analysis was structured around three dimensions: model family, model size, and prompt variation. For each task (correction and extraction), we tried to answer:

- Which model families perform best?
- How does model size affect performance-to-latency trade-offs?
- How much do prompt strategies improve outcomes?

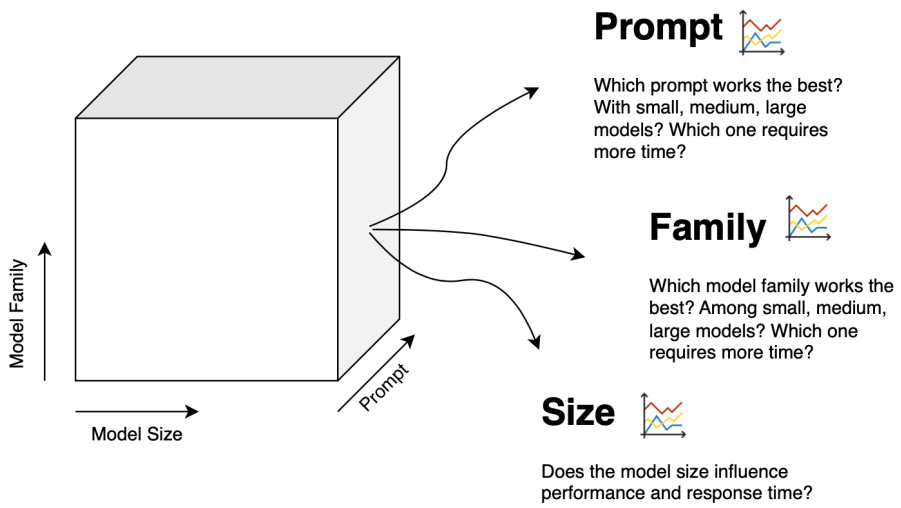


Figure 4.4: The three analysis dimensions: model family, model size, and prompt variation.

Rather than identifying a single "best" configuration, this approach provides actionable insights: users prioritizing speed can see how performance degrades with smaller models, while those prioritizing accuracy can identify the optimal family-size-prompt combination regardless of computational cost. The analysis will be discussed in Chapter 5.

4.2. Implementation

As stated in Chapter 2, models were constrained to open-source LLMs. We opted for the Ollama framework [15, 16], which facilitates the execution of quantized models on consumer-grade hardware.

The programming language chosen for this thesis is Python. The full project is available at: https://github.com/JohnCarta01/MyThesis. For privacy and data ownership concerns, the public repository contains exclusively the developed codes: the clinical data employed in this thesis are not available.

4.2.1. Models

For this study we selected open-source Large Language Models (LLMs) with sizes ranging from 4 to 70 billion parameters. This range was chosen to ensure compatibility with local hardware configurations providing between 4 and 40 GB of VRAM. To manage these models, we employed the Ollama framework [15, 16], which allows for the efficient execution of quantized models (using 4-bit quantization) on consumer-grade GPUs.

All models were executed using Ollama's default sampling parameters. The specific configuration is:

- **Temperature (0.8):** Controls the randomness of the output. A value of 0.8 allows for slightly creative responses while maintaining linguistic structure.
- **Top-P (0.9):** Limits the model's choices to a subset of tokens whose cumulative probability is 90%. It ensures that the model ignores the low-probability tokens.
- **Top-K (40):** Restricts the model to selecting from only the top 40 most likely next tokens. Like top-p, this prevents the model from choosing highly improbable tokens.

Preliminary tests involving temperature adjustments showed negligible performance gains compared to the impact of prompt modifications. Therefore, this work prioritizes prompt engineering as the primary variable, leaving fine-tuning of sampling parameters for future investigations.

Regarding the reliability of the results, we acknowledge the inherent challenges of reproducibility in LLM inference. While fixing random seeds is a standard practice, recent research by Yuan et al. (2025) [27] highlights that numerical non-determinism, caused by the non-associative nature of floating point arithmetic, can lead to divergent outputs

based on the specific GPU architecture. Consequently, we opted not to set a random seed in this study to avoid a false sense of determinism.

Our experiments tested models of seven distinct families available in Ollama's library [28]:

- **Gemma 3** (4B, 12B, 27B): Google's lightweight and efficient model series [29]
- **MedGemma** (4B, 27B): Medical domain specialized variant of Gemma3 models [30]
- **Llama 3** (8B, 70B): Meta's general-purpose model [31]
- **Mistral** (7B, 24B): Models optimized for high throughput and low latency [32, 33]
- **DeepSeek** (7B): An architecture optimized for complex reasoning and dialogue [34]
- **Qwen 3** (8B, 14B, 32B): Alibaba's multilingual models [35]
- **GPT-OSS** (20B): OpenAI's open weight models [36]

Model Family	Italian Support Level	Comment
Gemma 3	High	Native multilingual support from Google
MedGemma	High	Inherits Gemma's multilingual capabilities
Qwen 3	High	Officially supports 29+ languages, including Italian
Mistral	High	European-centric training, Latin languages focus
Llama 3	High	Extensive multilingual pre-training
DeepSeek	Moderate	More oriented to reasoning tasks
GPT-OSS	Emerging	General proficiency derived from open web datasets

Table 4.1: Italian language support across the selected model families

4.2.2. Correction Prompts

The correction task aims to transform noisy clinical shorthand into professional, expanded medical language while preserving all original information. We designed and tested correction prompts progressing from simple baselines to multi-strategy approaches [37].

All prompts were written in Italian to match the language of the notes. This choice was made to avoid introducing unnecessary linguistic noise and to better reflect real-world usage conditions.

Basic Italian Prompt (basic_ita)

This baseline prompt provides minimal guidance, explicitly assigning a role to the model while defining the task and output constraints. It relies on the model's pre-trained understanding of obstetric terminology to interpret the text.

```
Sei un AI addestrato per correggere ed espandere note manuali di
visite ostetriche.  Espandi le abbreviazioni secondo la terminologia
ostetrica e correggi gli errori di battitura.  Non aggiungere nuove
parole.  Non spiegare le correzioni.  Non cambiare i termini tecnici.

Nota da correggere:
{nota}
```

Which is in English:

```
You are an AI trained to correct and expand manually written notes
of obstetric visits.  Expand the abbreviations according to obstetric
terminology and correct the typographical errors.  Do not add new
words.  Do not explain the corrections.  Do not change the technical
terms.

Note to correct:
{note}
```

Few-Shot Prompt (few_shot)

Few-shot prompting is a well-established technique showing that 1-5 demonstrations of input-output pairs significantly improve model performance on structured tasks. Here, we provide two real anonymized examples from the clinical notes by appending them to the basic_ita prompt text:

```
{basic_ita_body}

Esempi:

‘04 obesità severa BMI 46 - Di Spiezio biometria -2w da inizio
gravidanza ...’
→
‘Obesità severa (BMI 46), Di Spiezio, biometria fetale -2 settimane
dall’inizio della gravidanza...’

’...’
→
’...’

Nota da correggere:
{nota}
```

Positive Instructions Prompt (only_positive)

Research on prompt engineering suggests that positive instructions (“do X”) may improve compliance compared to negative ones (“don’t do Y”) [37]. This variant reformulates constraints as actionable directives.

Basic prompt constraint	Positive prompt instruction
"Do not add new words" + "Do not change the technical terms"	"Keep the words of the original notes"
"Do not explain the corrections"	"Answer with only the correction"

Table 4.2: The evolution of constraints into positive directives

Common Acronyms Prompt (common_acronyms)

While modern LLMs often rely on external web searches or complex Retrieval-Augmented Generation (RAG) [38] to handle unknown terms, such capabilities are often unavailable or undesired in strictly offline and simplified environments. In this context, we emulate these retrieval behaviors by directly supplying the necessary domain-specific knowledge within the prompt.

More specifically, this prompt extends the Basic Italian one by appending a curated list of the most frequent obstetric abbreviations. This list was developed in collaboration with the medical expert as discussed in Section (3.2) and is provided in full in Appendix A.1.

```
{basic_ita_body}

Gli acronimi più comuni e le rispettive espansioni sono
- p.s.-> parto spontaneo, talvolta pronto soccorso
- t.c.-> taglio cesareo
- mef -> morte endouterina fetale
- ivg -> interruzione volontaria di gravidanza
- ...

Nota da correggere:
{nota}
```

Which is in English:

```
{basic_ita_body}

The most common acronyms and their respective expansions are
- ...

Note to correct:
{note}
```

Positive + Common Acronyms Prompt (pos_common_acronyms)

We combined positive instructions and domain-specific reference material to represent an approach between minimal guidance and comprehensive prompting.

```
{positive_instructions_body}

Gli acronimi più comuni e le rispettive espansioni sono
- ...

Nota da correggere:
{nota}
```

Complete Prompt (complete)

We finally integrated these techniques into a comprehensive prompt, employing a Structured Prompting strategy based on Google's guidelines for effective input design [37]. By organizing the prompt into distinct sections — such as role, context, and operational instructions — we established a clear information hierarchy that reduces ambiguity and enhances the model's ability to adhere to complex requirements. This structure is achieved through the use of Markdown delimiters, such as "#" to clearly separate functional blocks and "**" to provide typographical emphasis for key instructions [39, 40].

Here is reported the main structure of the prompt - translated in English. For the full Italian prompt and its English translation you can check Appendix A.3.1.

```
# ROLE
Act as an AI trained ...  Your task is to...

# CONTEXT AND GLOSSARY
Use the following glossary...:
{List of acronyms and expansions}

# OPERATIONAL INSTRUCTIONS
1.  **Expansion and Correction**:  Expand...
2.  **Preservation**:  Maintain...
3.  **Ambiguity Management**:  If an acronym is not present...
4.  **Integrity**:  Do not add new words...
5.  **Output Format**:  Respond exclusively with the corrected
note...

# EXAMPLES (FEW-SHOT)
{Three examples with input and output}

# NOTE TO CORRECT:
{Note}
```

Prompt Repetition

Inspired by the Google study published in December 2025, "Prompt Repetition Improves Non-Reasoning LLMs" [41], we implemented double and triple prompting strategies. We include a simplified illustrative example to convey the core idea, although the approach may appear overly simple at first glance:

Given a Naive prompt:

```
Correct this note:
{note}
```

The Double Naive prompt would be:

```
Correct this note:
{note}

Correct this note:
{note}
```

We did it with the Complete prompt and the Positive Common Acronyms prompt, both doubling and tripling them. This allowed us to check how well the technique would work in our context and with our prompts and tasks.

Prompts produced in this section:
Double Pos. + C.A. prompt
Double Complete prompt
Triple Pos. + C.A. prompt
Triple Complete prompt

Table 4.3: Repetition prompts tested for the correction task

List of prompts tested for correction task

The complete list of prompts employed for the Correction task is

- Basic Italian
- Few Shot
- Positive Instructions
- Common Acronyms
- Positive Common Acronyms
- Complete
- Double Positive Common Acronyms
- Double Complete
- Triple Positive Common Acronyms
- Triple Complete

4.2.3. Extraction Prompts

Extraction Fields and Pydantic Library

The extraction task requires models to identify and structure specific clinical data fields from unstructured text. The fields we asked the model to extract are the 28 individuated in the Qualitative Exploration section 3.2. Here we recall them, the full list with their descriptions is available in Appendix A.2:

1. **Administrative** (5 fields): Clinical code, department, physician name, delivery type, birth date
2. **Neonatal Information** (7 fields): Weight, length, head circumference, Apgar scores (separate and combined), gender
3. **Blood Gas and Metabolic Values** (10 fields): pH, pCO_2, pO_2, HCO_3 (standard and active), base excess variants, lactate, total CO_2
4. **Maternal Health** (5 fields): Pre-pregnancy weight, weight gain, hypertension status, diabetes status, obesity status
5. **Extra** (1 field): Information not captured by above categories

We implemented structured outputs via Ollama using the BaseModel solution provided by Pydantic library [42, 43]. This method allowed us to define a personalized output schema and enrich fields with specific descriptions: we built a BaseModel class *ValoriEstraibili* made of the 28 fields of our interest, each one with name and description. When calling the Ollama models, we only had to specify the output format citing the *ValoriEstraibili* class.

Algorithm 4.1 Ollama Call for Extraction

```
response = ollama.chat(
    ...,
    format = ValoriEstraibili.model_json_schema()
)
```

Basic Extraction Prompt (`basic`)

Like we did for Correction task, we started with a simple prompt. Since anything related to the fields and their descriptions was communicated to the model through the Pydantic method, we only included simple instructions about the task:

```
Estrai i valori perinatali presenti nella seguente nota.  Se un
valore NON è esplicitamente presente nella nota, restituisci una
stringa vuota per quel campo.  La nota è:
{nota}
```

Which is in English:

```
Extract the perinatal values present in the following note.  If a
value is NOT explicitly present in the note, return an empty string
for that field.  The note is:
{note}
```

Field Description Prompt (`field_desc`)

To evaluate if explicit context improves accuracy, we tested a strategy of including the field names and their descriptions into the prompt. To maintain consistency, we developed a function that converts the ValoriEstraibili schema metadata into a formatted string. This function is then called during prompt construction, ensuring that any updates to the Pydantic model are automatically reflected in the prompt.

Pseudo code of the function get_field_descriptions:

Algorithm 4.2 Get Fields Descriptions

```
descriptions = ""
for field in fields do
    desc = field[description]
    descriptions.append(field + ": " + desc)
    descriptions.append(newline)
end for
return descriptions
```

Prompt construction with the function call:

```
Estrai i valori perinatali presenti nella seguente nota.  Se un
valore NON è esplicitamente presente nella nota, restituisci una
stringa vuota per quel campo.

I campi da estrarre sono:
{get_fields_descriptions()}

La nota è:
{note}
```

Which is in English:

```
Extract the perinatal values present in the following note.  If a
value is NOT explicitly present in the note, return an empty string
for that field.

The fields to extact are:
{get_fields_descriptions()}

The note is:
{note}
```

Complete Prompt (complete)

We built a maximum-information prompt that provides role clarity, numbered operational rules, full JSON schema, explicit field-specific constraints, and a worked example demonstrating proper JSON structure and value formatting. Its full text is available in Appendix A.3.2, while here is reported the primary structure:

```
# ROLE
Act ...

# EXTRACTION INSTRUCTIONS
1. **Output Format**: ...
2. **Missing Data**: ...
3. **Precision**: ...
4. **Sex**: ...
5. **Apgar**: ...
6. **Hypertension/Diabetes/Obesity**: ...

# JSON SCHEMA (Fields to extract)
{get_fields_descriptions}

# EXAMPLE (FEW-SHOT)
Input: ...

Output:
{
"clinical_code": ...,
"ward": ...,
"doctor": ...,
...
"extra": ...
}

# CLINICAL NOTE TO PROCESS:
{note}
```

Repetition Prompts

As done with prompts for Correction task, we tried to implement the repetition approach [41]. Here we did it with Field Description and Complete prompts.

Prompts produced in this section:
Double Field Desc prompt
Double Complete prompt
Triple Field Desc prompt
Triple Complete prompt

Table 4.4: Repetition prompts tested for the extraction task

List of prompts tested for Extraction task

The complete list of prompts employed for the Extraction task is:

- Basic Italian
- Fields Descriptions
- Complete
- Double Field Descriptions Acronyms
- Double Complete
- Triple Fields Descriptions Acronyms
- Triple Complete

5 | Evaluation and Results

This chapter presents the comprehensive evaluation methodology and detailed results from both the note correction and structured field extraction tasks introduced in Chapter 2. It includes a description of the metrics used and a commentary on the findings of the tests, executed on the combinations of models and prompts described in Chapter 4.

5.1. Metrics

5.1.1. Metrics for Correction Task

Correction quality is evaluated using both string-based and semantic similarity metrics applied to the 100 evaluation notes of the test set. For the string-based evaluation we opted for word-based binary metrics and BLEU scores, while for the semantical evaluation we computed embedding similarities between LLMs corrections and target notes (often referenced in literature as BERTscore, since BERT models are commonly used for this task [44]).

Word-Level Binary Metrics

We evaluated the performance using "classical" binary metrics, a group of metrics whose name comes from the typical problem they are used to evaluate: binary classification, where an item is either correctly or incorrectly identified. In the context of NLP and text correction, these metrics treat words as discrete, "all-or-nothing" units.

A word is considered a:

- **True Positive (TP)** if it appears in both the model's output and the ground truth
- **False Positive (FP)** if it is generated by the model but does not exist in the reference
- **False Negative (FN)** if it exists in the reference but was omitted by the model
- **A note about True Negative (TN)** the "True Negative" class would count every

word in the entire vocabulary that the model correctly chose not to generate: for this reason, the TN class is not considered in this context

Actually, to reflect the frequency of errors rather than just the presence of unique words: the True Positive count for a given word is calculated as the minimum of its occurrences in the reference and the prediction, while False Positives and False Negatives represent the surplus counts in the prediction and reference, respectively.

Here we provide an example:

- **Prediction:** *newborn female newborn good conditions*
- **Reference:** *newborn female good general conditions*

Word	Ref Count	Pred Count	TP	FP	FN	Rationale
newborn	1	2	1	1	0	One match; one repetition
female	1	1	1	0	0	Exact match
good	1	1	1	0	0	Exact match
conditions	1	1	1	0	0	Exact match
general	1	0	0	0	1	Omitted from prediction
Total	**5**	**5**	**4**	**1**	**1**	

Table 5.1: Example of word-level metric calculation using token-frequency alignment.

Starting from TP, FP and FN values, the following metrics are calculated:

- **Accuracy**: Measures the overall proportion of matches to the total number of unique errors and matches identified in the comparison, differing from the traditional definition of accuracy used in binary classification for the missing of "True Negative" class.

$$Accuracy = \frac{TP}{TP + FP + FN} \tag{5.1}$$

- **Precision**: Quantifies the model's reliability by calculating the fraction of generated words that correctly match the ground truth reference.

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

- **Recall**: Quantifies the model's ability to capture the information originally present in the reference note.

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

- **F1-score**: Provides a balanced performance metric by calculating the harmonic mean of precision and recall, penalizing both hallucinations and omissions.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.4}$$

BLEU Scores

The BiLingual Evaluation Understudy (BLEU) is a standard metric used to evaluate the quality of machine-generated text by comparing it to a reference one [45]. It relies on n-gram precision, which measures the overlap of contiguous sequences of n words between the prediction and the ground truth. It also penalizes predictions that are shorter than reference texts.

$$BLEU_n = BP \times \exp\left(\sum_{i=1}^{n} \frac{1}{n} \log p_i\right) \tag{5.5}$$

where p_i is the precision for n-grams of size i, and BP is Brevity Penalty.

- **N-gram Precisions (p_i)**: Measure the accuracy of the model across different sequence lengths from p-1 (unigrams) evaluating individual word choice, to p-4 (4-grams) assessing local word order and fluency. The Precisions are calculated as described in Equation (5.2) and then their geometric mean is considered.
- **Brevity Penalty (BP)**: Intended to penalize short outputs, which may achieve high precision by matching a limited number of "safe" tokens rather than providing the full context. It ensures that the model is penalized for omitting information present in the reference.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{5.6}$$

  where c is the length of the candidate text and r is the length of the reference text.

Semantic Similarity

We employed semantic similarity measures based on vector embeddings: an embedding is a dense vector representation where the meaning of a word or sentence is encoded into a high-dimensional space. This allows the evaluation to recognize that terms like *"newborn"* and *"infant"* are semantically similar, despite having almost no characters in common.

To quantify semantical similarities, the BERT model or its variations are typically used [44]. For this reason, these metrics are refereed in the literature as BERTScore. BERTscores leverage the contextual embeddings to align tokens between the prediction and reference

based on their cosine similarity [44, 46]. This ensures that the model is rewarded for clinical accuracy even when it uses a valid synonym not present in the ground truth.

$$\text{Similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \tag{5.7}$$

where $\mathbf{u}$ and $\mathbf{v}$ represent the embedding vectors of the prediction and reference, respectively.

We used two specific models for computing embedding vectors to compare:

- **SapBERT**: Pre-trained on the Unified Medical Language System (UMLS), this multilingual BERT variant is capable of capturing domain-specific equivalence between specialized biomedical terms, effectively mapping varied clinical jargon to the same conceptual space. In particular, we used the version trained with UMLS 2020AB, using xlm-roberta-large as the base model [47].
- **mE5 (multilingual E5)**: This model is trained using a contrastive learning objective on a massive multilingual dataset. It was chosen for its state-of-the-art ability to generate robust text representations, providing a reliable measure of semantic overlap that is less sensitive to specific linguistic syntax or Italian-specific phrasing [48, 49].

5.1.2. Metrics for Extraction Task

We evaluated field extraction quality using three metric families. The evaluation distinguishes between whether fields are correctly identified as filled or empty (FOEF), whether extracted values match target values precisely (Exact Match), and whether values match after accounting for common formatting inconsistencies (Normalized). Each metric group follows the structure of binary classification, defining outcomes as True and False Positives, True and False Negatives, and metrics as Accuracy, Precision, Recall and F-1.

Full Or Empty Field (FOEF) Metrics

FOEF metrics evaluate the binary classification of whether a field should be filled or remain empty, independent of the actual extracted value correctness. This metric family is valuable for assessing a model's ability to recognize which values are present in a given note.

For FOEF evaluation, the confusion matrix is defined in terms of field presence/absence:

- **TP (True Positive):** Field is filled in target AND model predicted it as filled

- **FP (False Positive):** Field is empty in target BUT model predicted it as filled
- **FN (False Negative):** Field is filled in target BUT model predicted it as empty
- **TN (True Negative):** Field is empty in target AND model predicted it as empty

And consequently, the binary FOEF Metrics are defined as follows:

- **FOEF Accuracy:** Measures the overall proportion of fields with correctly identified presence/absence status across all evaluated fields:

$$\text{FOEF Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.8}$$

- **FOEF Precision:** Of all fields predicted as non-empty, the fraction that should indeed be filled:

$$\text{FOEF Precision} = \frac{TP}{TP + FP} \tag{5.9}$$

  High precision indicates conservative extraction behavior with few false positives (field hallucination).

- **FOEF Recall:** Of all fields that should be filled (target-filled), the fraction correctly identified as non-empty by the model:

$$\text{FOEF Recall} = \frac{TP}{TP + FN} \tag{5.10}$$

  High recall indicates comprehensive extraction with few missed fields (field omission).

- **FOEF F1 Score:** Harmonic mean providing balanced metric for field presence detection:

$$\text{FOEF F1} = 2 \times \frac{\text{FOEF Precision} \times \text{FOEF Recall}}{\text{FOEF Precision} + \text{FOEF Recall}} \tag{5.11}$$

Exact Match (EM) Metrics

These metrics are more stringent than FOEF, requiring exact string matching without any normalization or interpretation. For EM evaluation, the confusion matrix is defined as:

- **TP (True Positive):** Field filled in target AND model extracted value that exactly matches the target value
- **FP (False Positive):** Field filled by model BUT either empty in target OR value

differs from target

- **FN (False Negative):** Field filled in target BUT model left it empty OR extracted a different value

And the metrics are:

- **EM Accuracy:** Proportion of fields where predicted value exactly matches target (computed for all fields including empty):

$$\text{EM Accuracy} = \frac{\#\text{ fields with exact matching values}}{\text{total }\#\text{ fields}} \tag{5.12}$$

- **EM Precision:** Of all fields where the model extracted a value, the fraction with exact match to target:

$$\text{EM Precision} = \frac{TP}{TP + FP} \tag{5.13}$$

  Low EM precision indicates frequent value mismatches or hallucinations.

- **EM Recall:** Of all fields filled in the target, the fraction with exact match to the model's prediction:

$$\text{EM Recall} = \frac{TP}{TP + FN} \tag{5.14}$$

  Low EM recall indicates frequent value mismatches or omissions.

- **EM F1 Score:** Harmonic mean providing balanced metric for exact match evaluation:

$$\text{EM F1} = 2 \times \frac{\text{EM Precision} \times \text{EM Recall}}{\text{EM Precision} + \text{EM Recall}} \tag{5.15}$$

Normalized (NORM) Metrics

The Exact Match metric is particularly sensitive to formatting inconsistencies that do not represent extraction failures but rather differ in representation. Analysis of extraction outputs on the 100 evaluation notes revealed that approximately 22% of exact-match false negatives resulted from formatting variations rather than genuine extraction errors. Key formatting issues identified include:

- **Decimal separators**: Some notes have decimal numbers with comma $(7,28)$ and some others with period (7.28) for numeric values like pH and blood gas measurements; some models answer with one of them, some with the other, and some may use both of them in different answers

- **Range separators**: Apgar scores are separated by a hyphen (6-8) in target but

also by forward slash (6/8) in some model outputs

- **Abbreviations**: Some medical terms may be reported in the answer as abbreviated (e.g., *gestaz* vs. *gestazionale*, *iperten.* vs. *ipertensione*)
- **Prefix qualifiers**: Boolean fields prefixed with confirmation (e.g., "sì, gestazionale" vs. just "gestazionale")
- **Case sensitivity**: Text fields with different capitalization (e.g., "Cesarean" vs. "cesarean")

Before computing Normalized Metrics, we addressed these issues by implementing field-type-specific normalization before comparison:

- **NUMERIC fields** (pH, pCO_2, weight...): Normalize decimal separators by converting comma to period
- **APGAR field**: Normalize separators by converting forward slash to hyphen
- **BOOLEAN CATEGORICAL fields** (ipertensione, diabete): Expand abbreviations using domain-specific mappings; strip qualifiers; normalize yes/no/true/false variations
- **CATEGORICAL TEXT fields** (sesso, parto type): Case-insensitive comparison; normalize spacing

Normalized metrics follow the same structure as EM metrics (Accuracy, Precision, Recall, F1) but only after normalization has been applied. This provides a more accurate assessment of extraction quality by distinguishing between:

- **True extraction failures**: Model unable to identify or extract the field content
- **Formatting variations**: Model extracted correct information but in a different format than the target

5.2. Results and Discussion

For both of the tasks, we proceeded in a two-phase fashion: at first, we experimented with all the chosen models and the variants of Basic prompts. Then, only with the top-performing models, we tested Complete and Repetition prompts. This approach allowed us to evaluate the impact of Prompt Engineering techniques across model families and sizes, and subsequently attempt to maximize performance.

In Figure 5.1 we recall the analysis dimensions considered during the first phase.

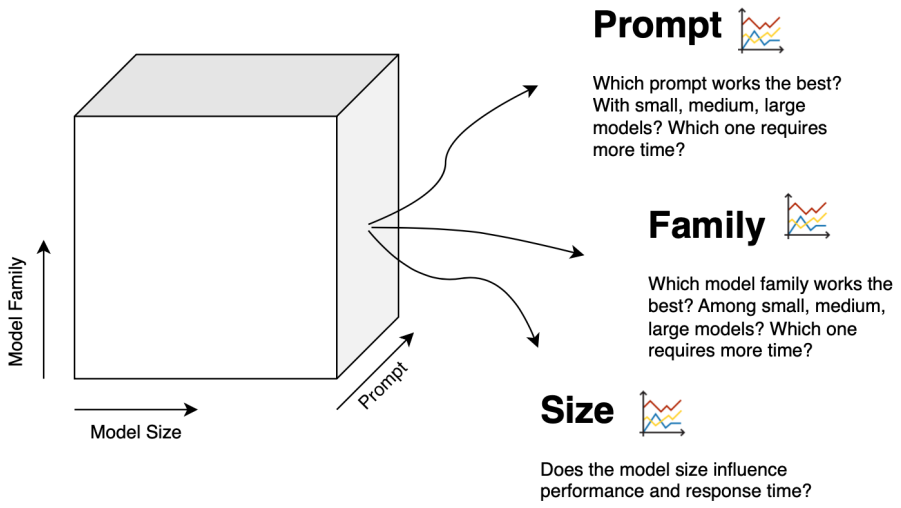


Figure 5.1: The three analysis dimensions: model family, model size, and prompt variation.

At first, we will discuss the Correction Task Results in Section 5.2.1, showing our findings across the three analysis dimensions and the top performing configurations. Then, we'll do the same for Extraction Task Results in Section 5.2.2.

As anticipated in Section 4.1.2, we implemented a bootstrapping with replacement procedure to compute 95% confidence intervals (CI) for all evaluation metrics. Consequently, the values reported in the following sections represent the means of means, calculated across 10,000 bootstrap iterations. Each iteration used a sample size of n = 100, matching the dimensions of the original test set. For visual clarity and readability in our plots, confidence intervals are displayed symmetrically, with the exception of response time figures, where we present the actual, observed intervals.

5.2.1. Correction Task Results

We evaluated note corrections on 100 Italian medical notes using string-based and embedding-based metrics across multiple model families, sizes, and prompt configurations. During a first phase, we evaluated five prompt variants derived from refinements to the basic Italian prompt: **Basic Italian** (baseline), **Few-Shot** (with examples), **Only-Positive** (negative instructions removed), **Common Acronyms** (with obstetric acronyms list), and **Positive + Common Acronyms** (combined approach). Each prompt was evaluated across all 14 models. Subsequently, we tasked a subset of the best-performing models with: **Complete**, **Double** and **Triple Positive-Common-Acronyms**, and **Double** and **Triple Complete** prompts.

Among all the metrics detailed in the previous section, our discussion primarily focuses on:

- **F1-score:** We consider it the most representative of the Word-Based Binary Metrics. While Precision and Recall are crucial for fine-tuning - revealing whether a model is over-predictive or overly conservative - the global F1-score provides a single measure of overall correctness. Furthermore, precision plays a distinct role in BLEU scores.
- **BLEU-2:** Given that the notes are often very short, noisy, and composed of independent concepts, we considered BLEU-2 the optimal compromise. It bridges the gap between BLEU-1, which evaluates tasks where structure is irrelevant, and BLEU-3 or BLEU-4, which are preferred when the fluency of the prediction is a priority.
- **SapBERT Embedding similarity:** We observed that SapBERT and mE5 similarities were highly correlated. Although their absolute values differed (SapBERT similarities typically ranged between 80% and 90%, while mE5 were usually above 90%), they almost consistently agreed on which target-prediction pairs were semantically closer.

Prompt Analysis

Here we present, for each group of metrics, how each prompting technique influenced performance relative to the baseline Basic prompt. The scores displayed are the average scores for each prompt across all 14 models.

In these first plots, every metric is reported to illustrate the correlation within each group.

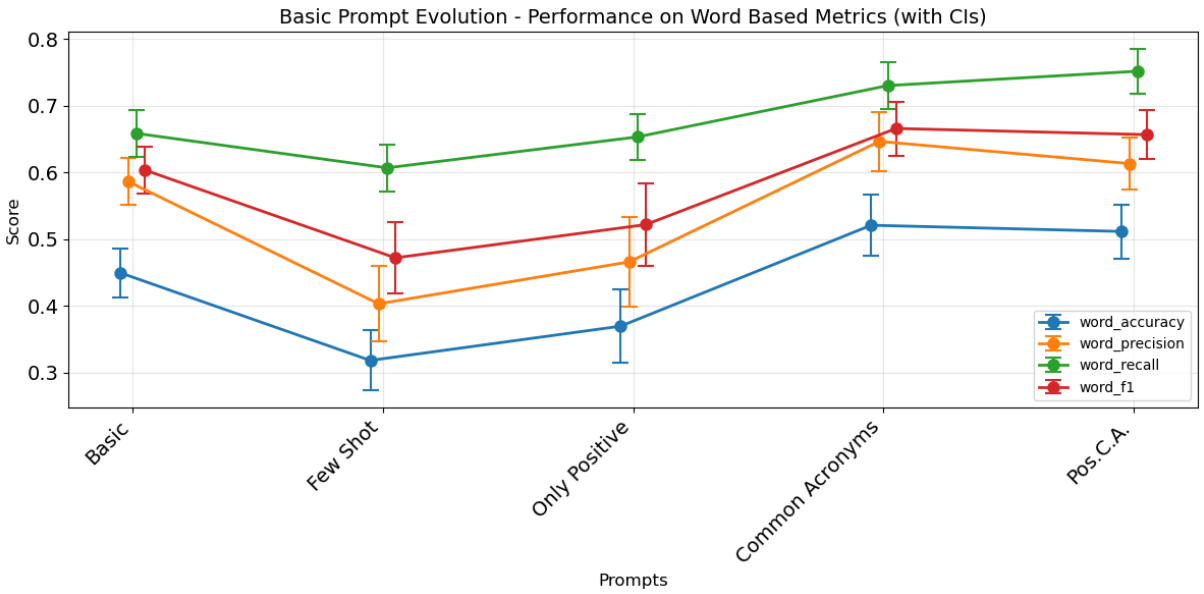


Figure 5.2: Word-based metrics and confidence intervals for performance of Basic, Few-Shot, Only Positive, Common Acronyms, and Positive Common Acronyms prompts on the correction task

As shown in Figure 5.2, the **Few-Shot** and **Only-Positive** strategies yield lower F1-scores compared to the baseline. Specifically, the **Only-Positive** prompt results in higher Recall but lower Precision, suggesting that removing negative constraints reduces False Negatives (omissions) but increases False Positives (hallucinations). However, when the **Only-Positive** strategy is combined with the **Common Acronyms** list, it does not negatively impact performance. This suggests that while positive-only instructions leave some ambiguity, providing domain-specific knowledge (acronyms) compensates for this, effectively guiding the model. The **Common Acronyms** prompt and its combination with Positive instructions achieve the best overall results in terms of F1-score.

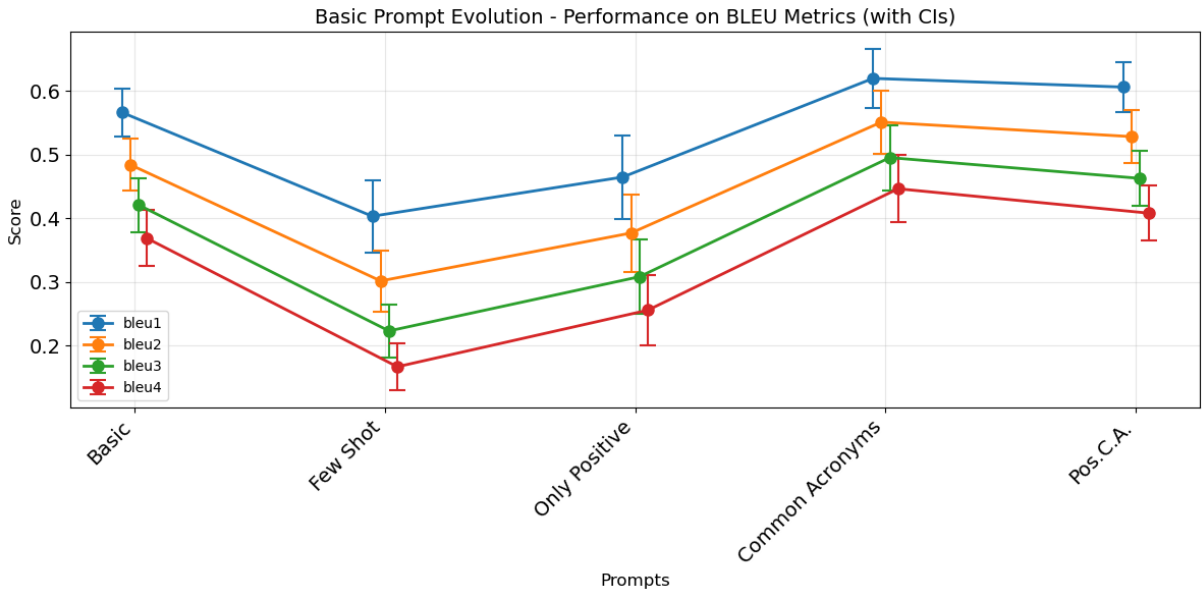


Figure 5.3: BLEU score performance with confidence intervals for Basic, Few-Shot, Only Positive, Common Acronyms, and Positive Common Acronyms prompts on the correction task

The analysis of BLEU scores, presented in Figure 5.3, confirms the trends observed with word-based metrics. The trend remains consistent across different n-gram lengths, with all BLEU scores showing a similar progression. This alignment suggests that the lower scores for BLEU-3 and BLEU-4 are primarily due to the same token-level errors identified by BLEU-1 and BLEU-2, rather than structural or ordering issues in the generated sentences.

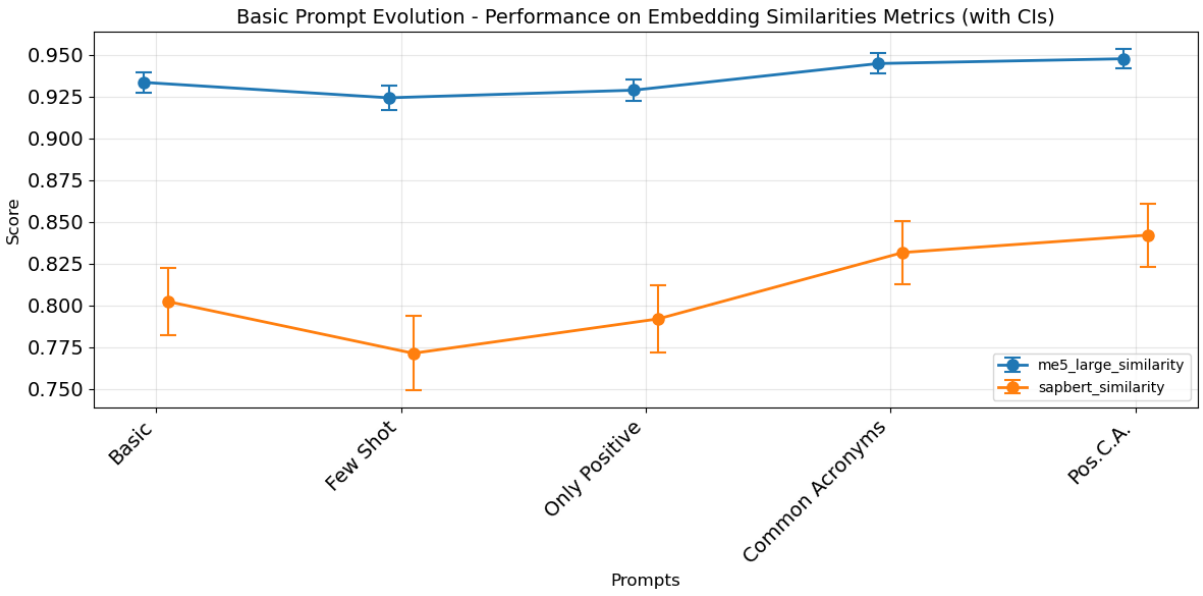
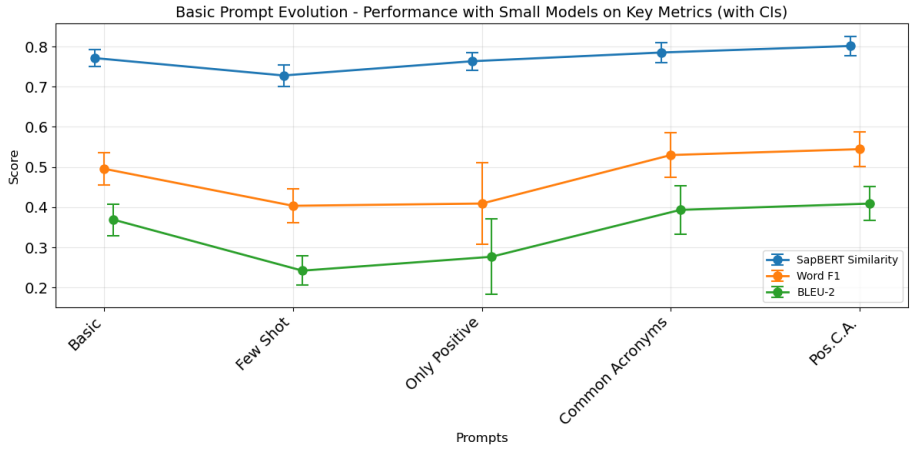


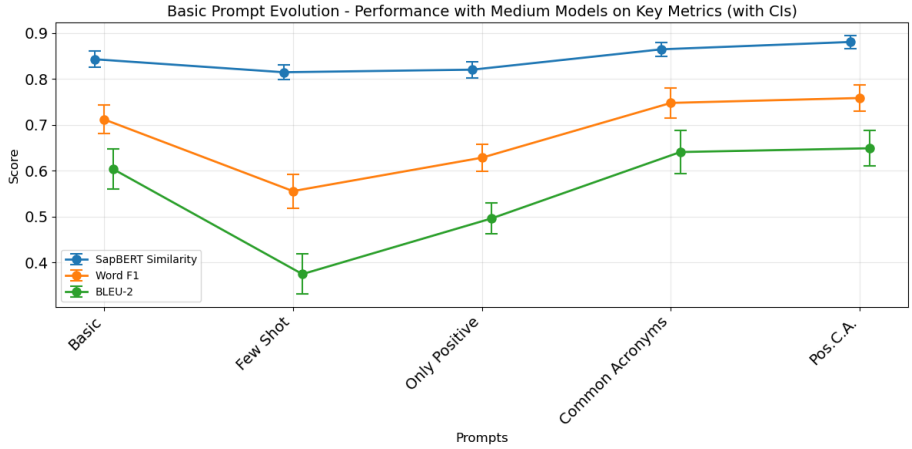
Figure 5.4: Embedding similarity and confidence intervals for performance of Basic, Few-Shot, Only Positive, Common Acronyms, and Positive Common Acronyms prompts on the correction task

Finally, Figure 5.4 demonstrates that embedding similarities strongly agree with previous metrics. Both SapBERT and mE5 show similar trends, with SapBERT exhibiting a wider variance that highlights the differences between prompts more clearly. Consistent with prior observations, prompts incorporating the Common Acronyms list achieve the highest semantic fidelity to the ground truth. Notably, in terms of semantic similarity, the combined Positive + Common Acronyms approach marginally outperforms the standalone Common Acronyms prompt.

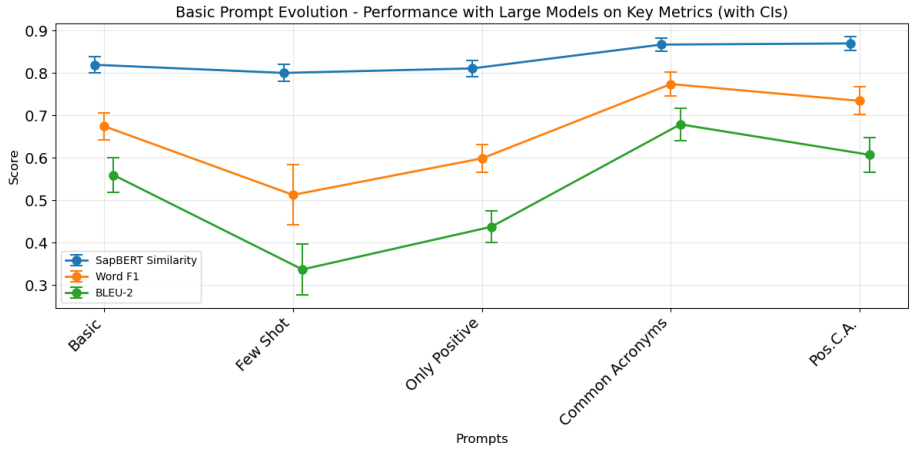
We broke down the performance by model size class to verify the consistent superiority of the Common Acronyms-based prompts:



(a) Correction Task: Prompts Performance with Small Models



(b) Correction Task: Prompts Performance with Medium Models



(c) Correction Task: Prompts Performance with Large Models

Figure 5.5: Key metrics and confidence intervals for performance with Small, Medium and Large models of Basic, Few-Shot, Only Positive, Common Acronyms, and Positive Common Acronyms prompts on the correction task

As illustrated in Figure 5.5, the performance gains provided by domain-specific context (acronyms) are universal across all model scales. These results refute the hypothesis that smaller models might be confused by prompt complexity; on the contrary, their mean scores improved significantly with additional context, mirroring the behavior of their larger counterparts.

A singular divergence in this trend occurs in larger models, where the **Positive-Common-Acronyms** configuration shows a decrease in performance on string-based metrics compared to **Common Acronyms** alone. Notably, this same configuration achieves superior results in embedding similarity, suggesting a shift toward higher semantic accuracy even when strict character-level alignment decreases. Given that both metric types are considered equally valuable in this evaluation, this shift highlights a trade-off between lexical precision and conceptual mapping in larger architectures.

Finally, we evaluated the impact of prompting on computational efficiency. Figure 5.6 shows the average response time for each prompt configuration.

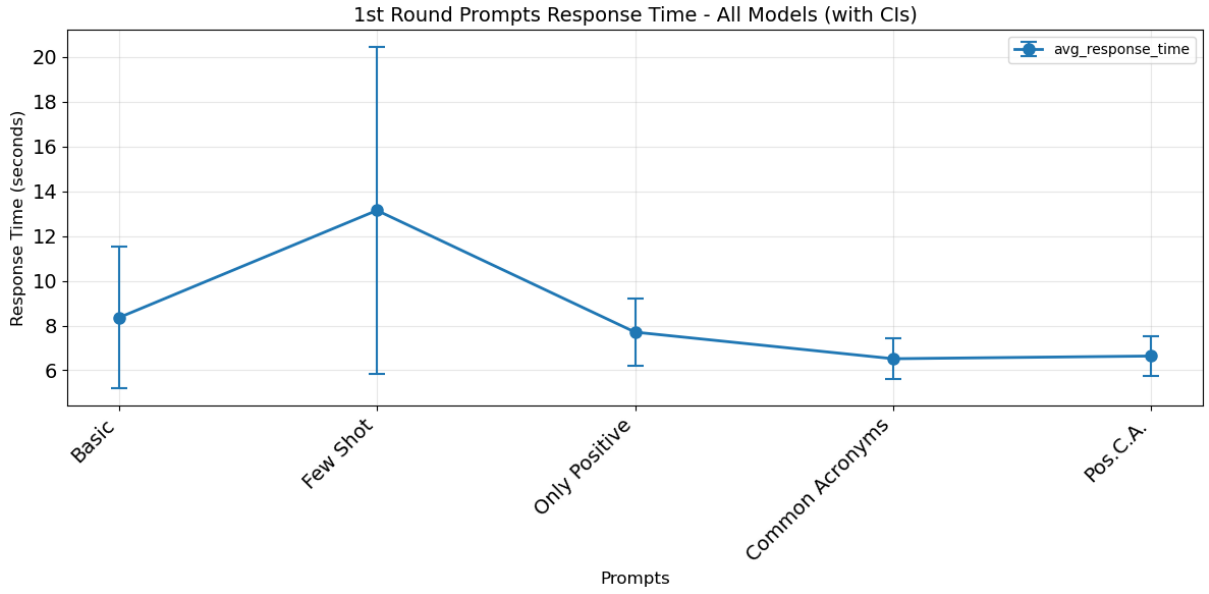


Figure 5.6: Response time and confidence intervals of Basic, Few-Shot, Only Positive, Common Acronyms, and Positive Common Acronyms prompts on the correction task

The **Few-Shot** prompt proves to be the most computationally expensive. Interestingly, apart from Few-Shot, providing more explicit directions and suggestions (as in the Acronyms prompts) appears to help the models converge to an output slightly faster, saving inference time while improving accuracy.

Model Size Analysis

In this subsection, we analyze how scaling model size affects performance. For this analysis, we exclude the GPT-OSS and DeepSeek families, as they were only tested with a single model size. Furthermore, considering the clear superiority of the **Common Acronyms** and **Positive + Common Acronyms** prompts demonstrated in the previous section, we report the average scores obtained using only these two prompts.

Table 5.2 and Table 5.3 present the results for families with two variants (Small/Large) and three variants (Small/Medium/Large), respectively. Only F1-score and BLEU-2 are reported for readability, but the trends are consistent across all metrics.

Family	**F1 Score**		**BLEU-2 Score**		**Improvement**	
	Small	**Large**	**Small**	**Large**	**F1 Gain**	**BLEU-2 Gain**
Llama3	0.499	0.613	0.311	0.439	+11.4%	+12.8%
Mistral	0.334	0.813	0.176	0.736	+47.9%	+56.0%
Medgemma	0.721	0.731	0.601	0.597	+1.0%	-0.4%

Table 5.2: Performance evolution (from *small* to *large* models) for Llama 3, Mistral and Medgemma families, measured on Word Based F1 score and BLEU-2 score

Family	**F1 Score**			**BLEU-2 Score**		
	Small	**Medium**	**Large**	**Small**	**Medium**	**Large**
Gemma3	0.700	0.721	0.771	0.579	0.598	0.663
Qwen3	0.725	0.785	0.810	0.618	0.692	0.726

Table 5.3: Performance evolution (from *small* to *medium* to *large* models) for Gemma 3 and Qwen 3 families, measured on Word Based F1 score and BLEU-2 score

The data indicates a positive trend between model size and performance, although the magnitude of improvement varies significantly by family. **Mistral** shows the largest gain, leading from poor performance in the small version to the best results in the large one. **MedGemma**, conversely, shows negligible improvement, suggesting that its domain-specific fine-tuning might already maximize the potential of the smaller architecture, or that the larger architecture does not add significant value for this specific task.

In Table 5.4 we show the same evolution but measured with SapBERT similarity:

Family	Size	SapBERT Similarity
Llama3	Small	0.770
	Large	0.810
Mistral	Small	0.751
	Large	0.878
MedGemma	Small	0.854
	Large	0.873
Gemma3	Small	0.854
	Medium	0.865
	Large	0.877
Qwen3	Small	0.861
	Medium	0.880
	Large	0.889

Table 5.4: Performance evolution for Llama 3, Mistral, Medgemma, Gemma 3 and Qwen 3 families, measured on SapBERT similarity

Trends are confirmed by the SapBERT similarity comparison: larger models outperform their smaller counterparts, sometimes with a small improvement (as in Medgemma family, for example) and sometimes with a significant one (e.g. Mistral family).

Regarding time efficiency, the goal is to show the correlation between model size and inference time, not to compare the absolute times of different families, which will be discussed in the next section. The first plot shows the full range of response times, while the second one is zoomed in to better visualize the differences among models which are not in **Qwen 3** family:

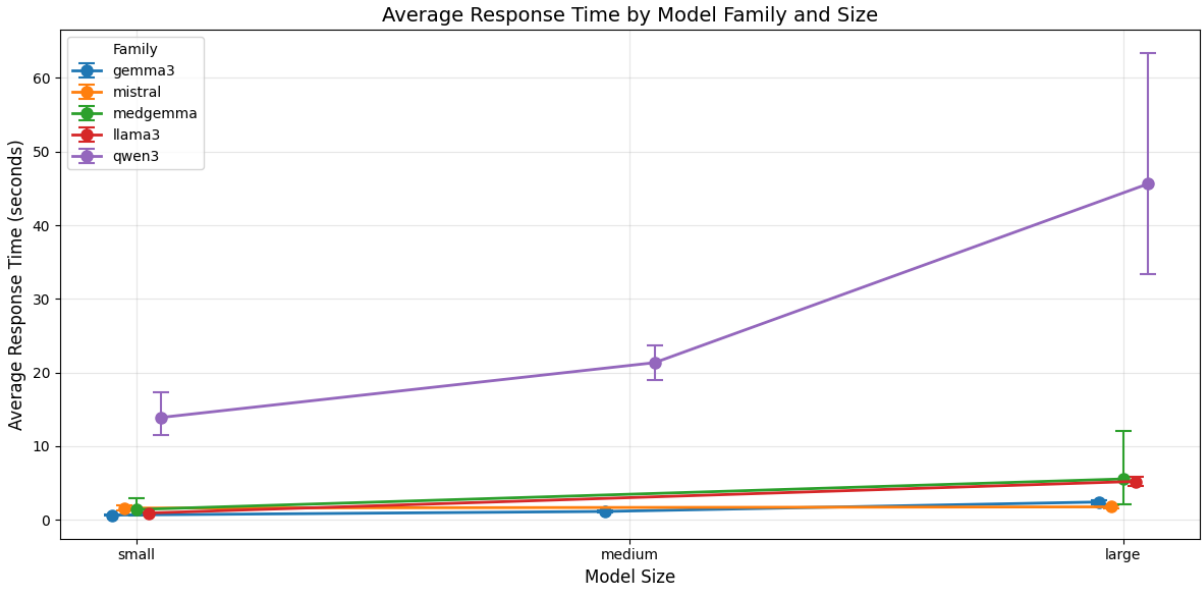


Figure 5.7: Response time and confidence intervals of Gemma 3, Mistral, Medgemma, Llama3, Qwen 3 families by model size on the correction task

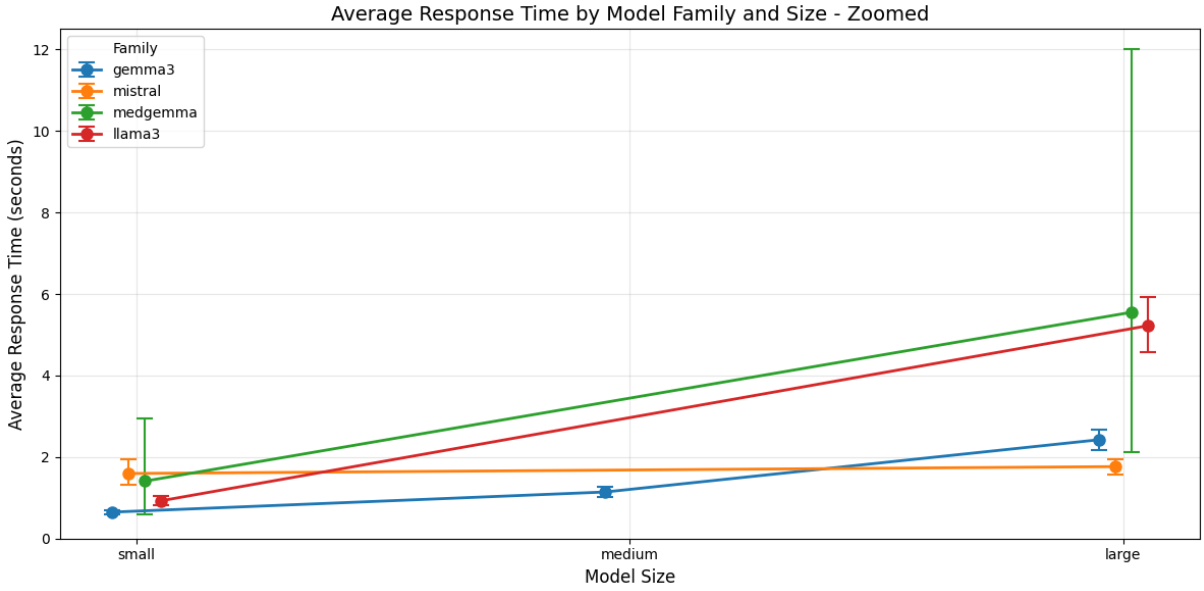


Figure 5.8: Response time and confidence intervals of Gemma 3, Mistral, Medgemma, Llama3 families by model size on the correction task

As expected, larger models require significantly more time to infer. Again, the only exception is represented by **Mistral** family.

Model Family Analysis

In this subsection, we compare model families, evaluating which architectures achieved superior correction performance when compared with opponents of similar size.

Table 5.5 ranks model families by performance on small models (4-8B parameters). We observe that **Qwen3** is the top performer in the small category.

Family	BLEU-2	Word F1	SapBERT Similarity
DeepSeek	0.123	0.245	0.670
Gemma3	0.579	0.700	0.854
Llama3	0.311	0.499	0.770
MedGemma	0.601	0.721	0.854
Mistral	0.176	0.334	0.751
Qwen3	0.618	0.725	0.861

Table 5.5: BLEU-2, Word Based F1 and SapBERT Similarity scores of *small* models tested on the correction task

For medium-sized models (10-15B parameters), Table 5.6 shows that **Qwen3** maintains its lead over Gemma3 across all metrics.

Family	BLEU-2	Word F1	SapBERT Similarity
Gemma3	0.598	0.721	0.865
Qwen3	0.692	0.785	0.880

Table 5.6: BLEU-2, Word Based F1 and SapBERT Similarity scores of *medium* models tested on the correction task

In the large category (20B+ parameters), as shown in Table 5.7, **Mistral** (Large) and **Qwen3** (Large) emerge as the clear leaders. Mistral, in particular, demonstrates exceptional performance, effectively matching Qwen3's F1 score and achieving the highest BLEU-2 score. **MedGemma**'s competitive performance in the small category is not confirmed in the large one.

Family	BLEU-2	Word F1	SapBERT Similarity
Gemma3	0.663	0.771	0.877
GPT-OSS	0.699	0.787	0.885
Llama3	0.439	0.613	0.810
MedGemma	0.597	0.731	0.873
Mistral	0.736	0.813	0.878
Qwen3	0.726	0.810	0.889

Table 5.7: BLEU-2, Word Based F1 and SapBERT Similarity scores of *large* models tested on the correction task

However, performance must be balanced against efficiency. Figure 5.9, Table 5.8 and Figure 5.10 reveal a crucial trade-off. While **Qwen 3** family offers top-tier accuracy, it is remarkably slower than its competitors. **Gemma 3** models, instead, stand out as extremely efficient, offering strong performance at a fraction of the inference time.

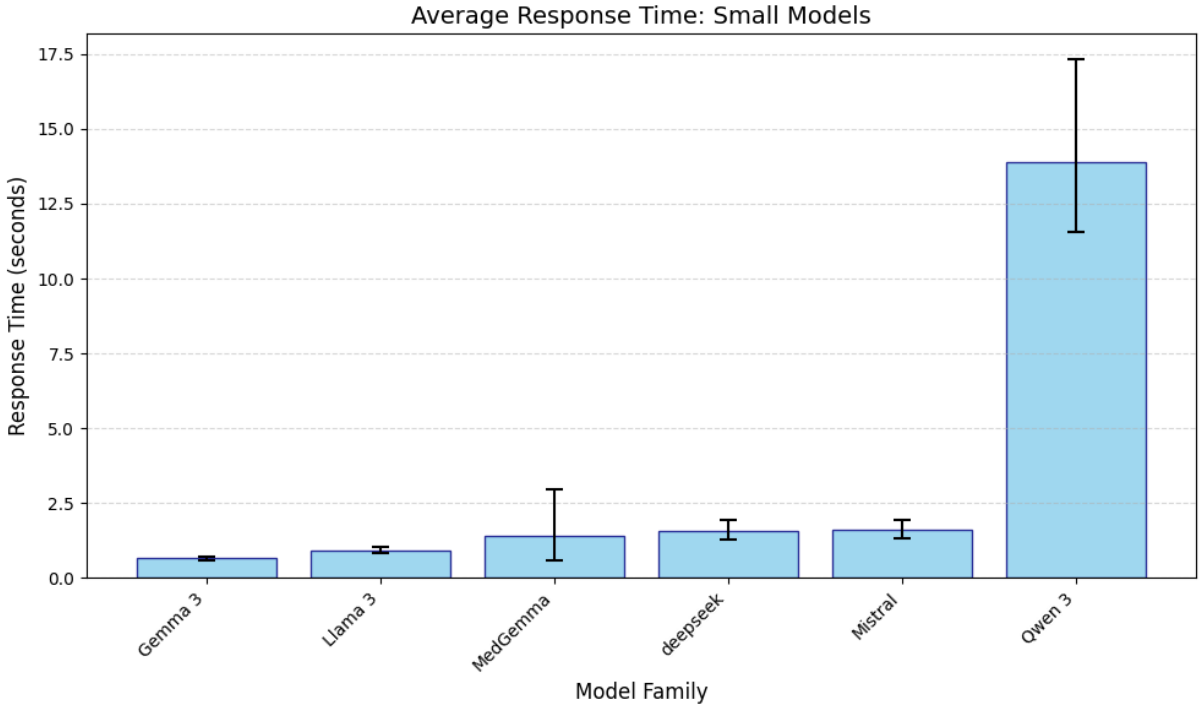


Figure 5.9: Response time and confidence intervals of the considered *small* models on the correction task

Model Family	Size	Avg. Response Time (s)
Gemma3	Medium	1.142
Qwen3	Medium	21.342

Table 5.8: Response time of the considered *medium* models on the correction task

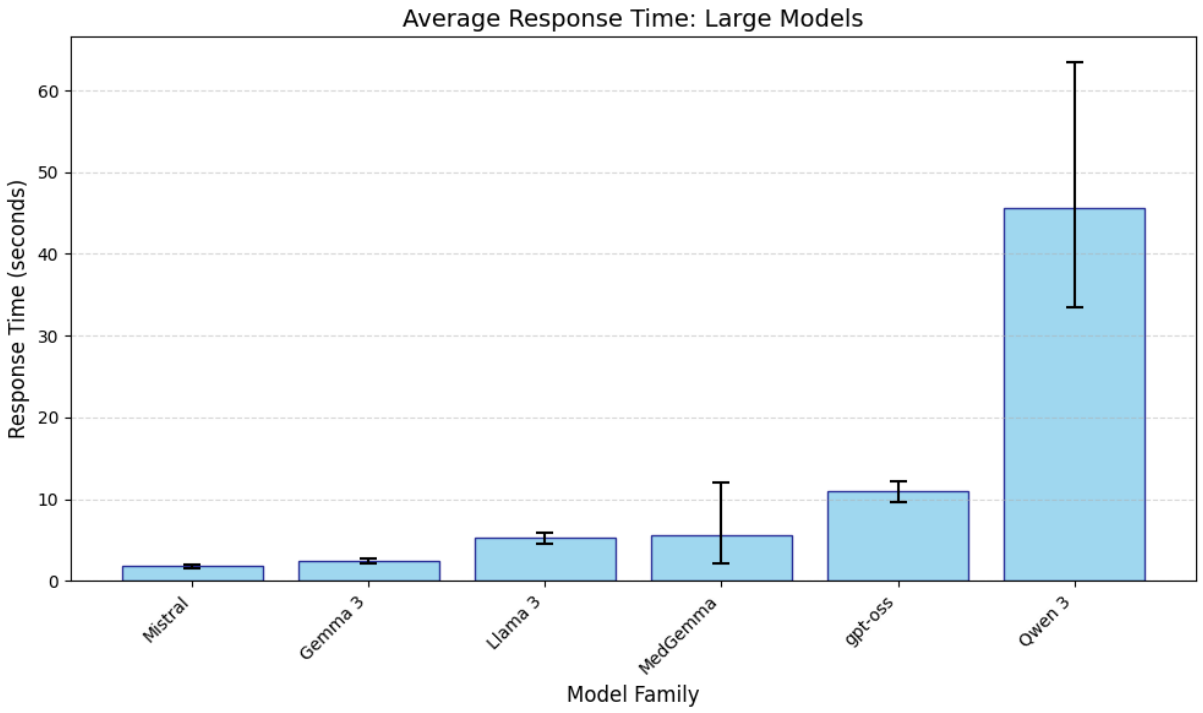


Figure 5.10: Response time and confidence intervals of the considered *large* models on the correction task

Advanced Prompting and Best Configuration

After identifying top-performing models in the Basic Prompt Evolution analysis, we applied advanced prompt engineering techniques to a subset of the models. These experiments included the utilization of the **Complete** prompt and the **Repetition** of prompts with the best performing models.

The Complete prompt was tested with Qwen Small and Large models derived from the previous phase. Table 5.9 compares the performance of the two best prompts from the first phase (the two with Common Acronyms) against the Complete prompt. Interestingly, the simpler Common Acronyms prompt yielded higher Word F1 and BLEU scores, while the Complete prompt and Positive + Common Acronyms prompt achieved the highest

semantic similarity (SapBERT), suggesting that the Complete prompt may generate more semantically rich but slightly less structurally precise corrections.

Prompt	BLEU-2	Word F1	SapBERT Similarity
Common Acronyms	**0.659**	**0.758**	0.863
P. + C.A.	0.635	0.748	**0.870**
Complete	0.613	0.734	**0.870**

Table 5.9: BLEU-2, Word Based F1 and SapBERT Similarity scores of Common Acronyms, Positive Common Acronyms and Complete prompts

Finally, we implemented the **Prompt Repetition** technique. We selected for this phase the two best non-reasoning models: **Mistral Large** and **Gemma3 Large**. We prioritized these over Qwen3 Large despite its high performance because of its prohibitive response time. Furthermore, Qwen3 is a "reasoning family" and the Prompt Repetition technique was originally validated on non-reasoning models [41].

Table 5.10 reports the mean of the scores obtained with the two models for each prompt configuration. The results show a clear trend: repeating the prompt (Double, Triple) consistently improves all metrics. The **Triple Positive + Common Acronyms** prompt achieved the highest F1 score (0.808) and BLEU-2 score (0.712). The **Triple Complete** prompt reached the highest SapBERT similarity (0.887).

Prompt	BLEU-2	Word F1	SapBERT Similarity
P. + C.A.	0.685	0.789	0.884
Double P. + C.A.	0.691	0.796	0.880
Triple P. + C.A.	0.712	0.808	0.884
Complete	0.633	0.752	0.874
Double Complete	0.689	0.788	0.884
Triple Complete	0.699	0.795	0.887

Table 5.10: BLEU-2, Word Based F1 and SapBERT Similarity scores for each correction task prompt; only scores of Mistral and Gemma 3 *large* models were considered

The trade-off for this improvement is increased inference time. Figure 5.11 shows the average response times, which were computed by averaging the response times *only* of Mistral and Gemma3 large models for each prompt configuration.

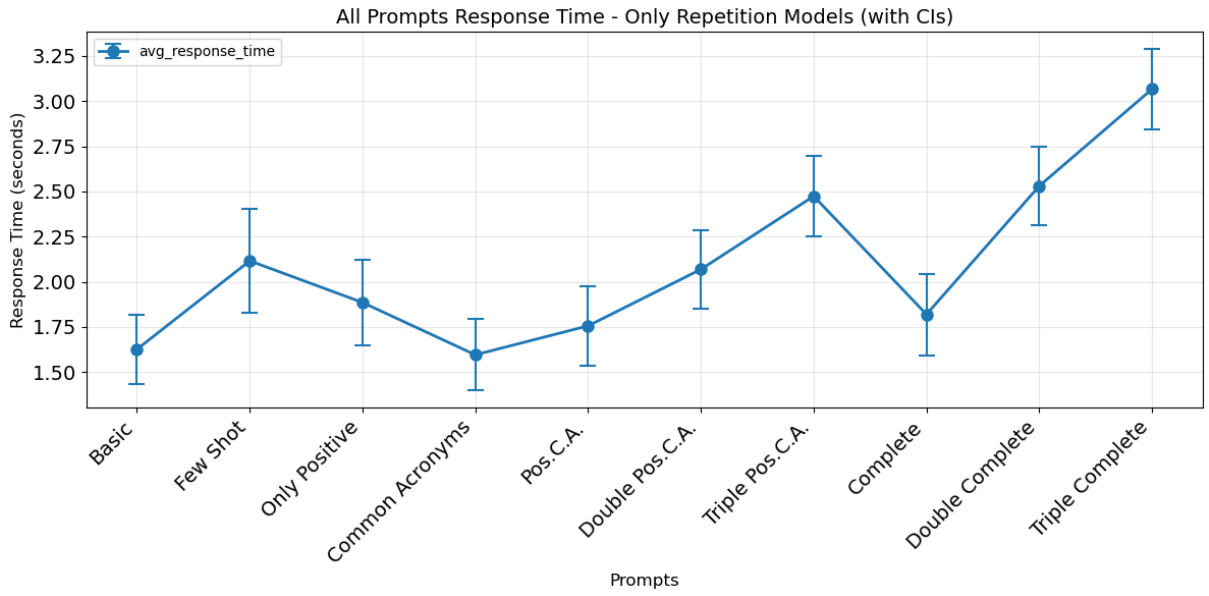


Figure 5.11: Response time and confidence intervals for each correction task prompt; only times of Mistral and Gemma 3 *large* models were considered

It is shown how the response time increases with the number of repetitions, with the Triple prompts being almost twice as slow as the single prompts.

Correction Task Findings

The experiment demonstrated that locally deployable models can be considered a promising solution to clean and correct clinical shorthands. The most significant finding is the power of domain-specific prompting: the simple addition of a list of common obstetric acronyms yielded larger performance gains than increasing model size or using few-shot examples. This is a critical insight for resource-constrained clinical settings, suggesting that "knowledge injection" via prompts is a viable alternative to computationally expensive fine-tuning.

Our response time analysis revealed a clear stratification. While models like Qwen3 offer top-tier accuracy, their high latency makes them less suitable for large datasets processing and unusable in real-time interaction scenarios. Conversely, the Gemma3 and Mistral families offer a convenient balance, providing competitive performance with significantly lower latency.

5.2.2. Extraction Task Results

We evaluated structured field extraction from Italian medical notes across 42 model-prompt configurations using FOEF (Full Or Empty Field), Exact Match, and Normalized metrics. We proceeded with a two-phase methodology as in the Correction Task.

To summarize the performance effectively, we report the F1 scores for each of the three metric groups: FOEF, Exact Match, and Normalized.

Prompt Analysis

For the extraction task, we experimented with a single major refinement on top of the Basic prompt before moving to the Complete one: incorporating a description of the output schema directly into the prompt string 4.2.3. This subsection discusses the improvements yielded by the **Field Description** prompt.

As shown in Table 5.11, adding the field descriptions significantly improves the **FOEF F1** (from 0.568 to 0.763), indicating that the model becomes distinctly better at locating the relevant information within the noise. Interestingly, while the Normalized F1 score improves, the strict Exact Match F1 score actually decreases. This phenomenon suggests that while the descriptions help the model correctly identify *what* to extract (improving FOEF and Normalized), they may also induce the model to "interpret" or "standardize" the values (e.g., converting units or changing separators) rather than performing a verbatim extraction, leading to mismatched strings in the strict metric.

Prompt	FOEF F1	Exact F1	Norm F1
Basic Italian	0.568	0.305	0.397
Field Description	0.763	0.246	0.442

Table 5.11: FOEF, Exact and Normalized F1 scores of Basic Italian and Field Description prompts on extraction task

We analyzed whether this improvement depends on model size. Table 5.12 breaks down the results for Small, Medium, and Large models.

Model Size	Prompt	FOEF F1	Exact F1	Norm F1
Small	Basic Prompt	0.544	**0.270**	0.370
	Field Description	**0.683**	0.228	**0.433**
Medium	Basic Prompt	0.706	**0.456**	**0.530**
	Field Description	**0.832**	0.275	0.444
Large	Basic Prompt	0.546	**0.289**	0.379
	Field Description	**0.820**	0.253	**0.426**

Table 5.12: FOEF, Exact and Normalized F1 scores of Basic Italian and Field Description prompts on extraction task, for different model sizes

The improvement in FOEF F1 is consistent across all size classes, as well as the Exact F1 score worsening. Instead, Normalized F1 shows an exception with medium models.

Surprisingly, the Field Description prompt actually decreases the average response time, as shown in Table 5.13. This suggests that providing a schema allows the models to generate the structured JSON output more directly.

Prompt	Average Response Time (s)
Basic Prompt	16.532
Field Description	**13.299**

Table 5.13: Response time of Basic Italian and Field Description prompts on extraction task

Model Size Analysis

Contrarily to the Correction Task, the Extraction Task does not show a clear linear correlation between model size and performance. Figure 5.12 illustrates that in some cases, smaller models perform comparably to or even better than their larger counterparts.

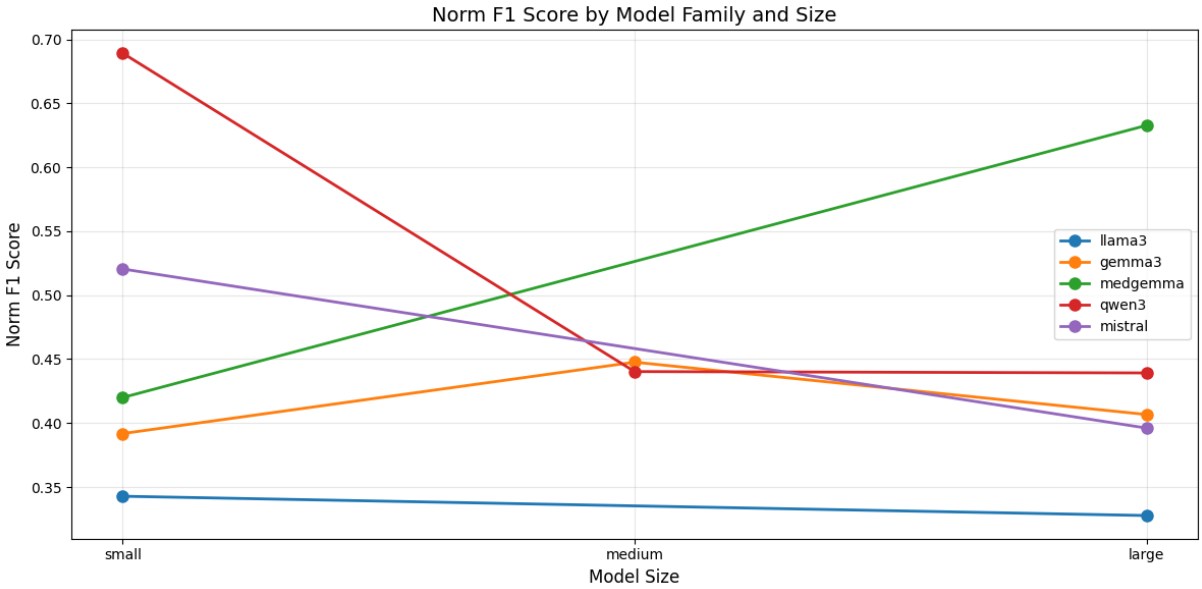


Figure 5.12: Normalized F1 score and confidence intervals of Llama 3, Gemma 3, Medgemma, Qwen 3, Mistral families by model size on the extraction task

We analyzed this phenomenon family by family:

For **Llama3** (Table 5.14), the small model outperforms the large one in both FOEF, Exact Match and Normalized Match:

Size	FOEF F1	Exact F1	Norm F1
Small	0.805	0.119	0.343
Large	0.794	0.080	0.328

Table 5.14: Performance evolution for Llama 3 family on extraction task, measured on FOEF, Exact and Normalized F1 scores

Gemma3 (Table 5.15) shows a "sweet spot" with the Medium model for Normalized F1, although FOEF scales with size.

Size	FOEF F1	Exact F1	Norm F1
Small	0.697	0.236	0.392
Medium	0.778	**0.288**	**0.448**
Large	**0.861**	0.264	0.407

Table 5.15: Performance evolution for Gemma 3 family on extraction task, measured on FOEF, Exact and Normalized F1 scores

For **Medgemma** (Table 5.16), the large model outperforms by a lot the small one in every F1 score.

Size	FOEF F1	Exact F1	Norm F1
Small	0.569	0.134	0.420
Large	0.754	0.476	0.633

Table 5.16: Performance evolution for Medgemma family on extraction task, measured on FOEF, Exact and Normalized F1 scores

Qwen3 (Table 5.17) shows unexpectedly high performance with its Small model, which practically matches or exceeds the Large model's results.

Size	FOEF F1	Exact F1	Norm F1
Small	0.877	**0.398**	**0.689**
Medium	0.886	0.261	0.440
Large	**0.901**	0.238	0.439

Table 5.17: Performance evolution for Qwen 3 family on extraction task, measured on FOEF, Exact and Normalized F1 scores

Finally, **Mistral** (Table 5.18) shows improvements only in FOEF F1 when upscaling.

Size	FOEF F1	Exact F1	Norm F1
Small	0.691	**0.299**	**0.520**
Large	**0.833**	0.238	0.396

Table 5.18: Performance evolution for Mistral 3 family on extraction task, measured on FOEF, Exact and Normalized F1 scores

This indicates that for structured extraction schema adherence, model architecture and training data may be more critical than pure parameter count.

The expected scaling trend is, however, still visible in the response time, with the only exception of **Qwen3 Small** model having an average response time higher than its medium counterpart.

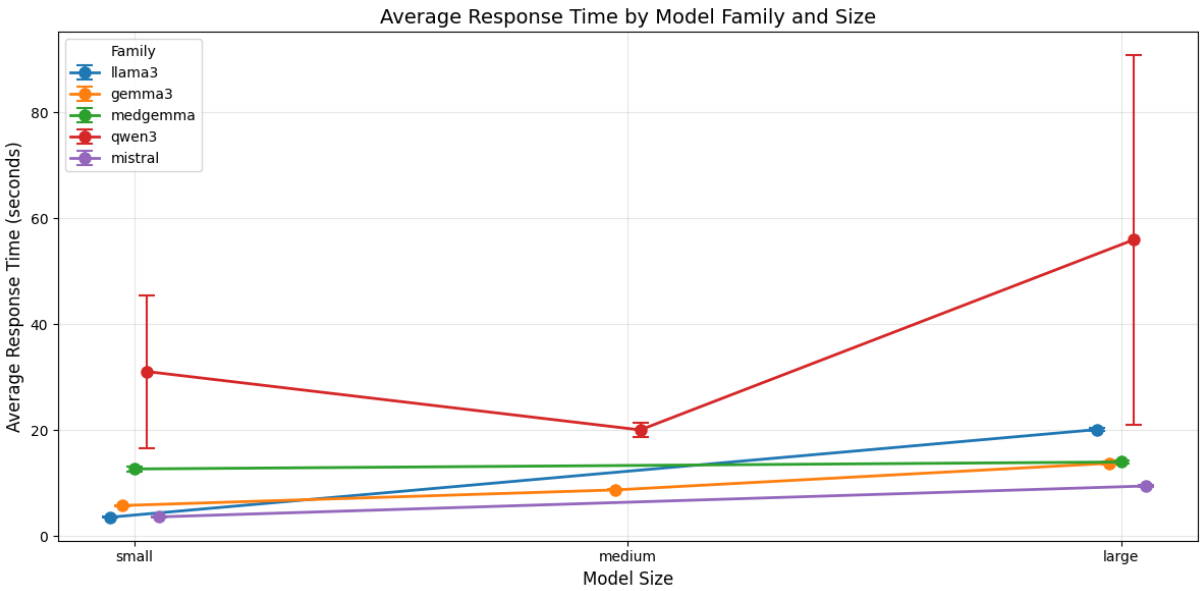


Figure 5.13: Response time and confidence intervals of Llama 3, Gemma 3, Medgemma, Qwen 3, Mistral families by model size on the extraction task

Model Family Analysis

We cross-referenced model families within each size bracket.

In the **Small** category (Table 5.19), **Qwen3** is the winner, achieving a Norm F1 of 0.689 and outperforming the second best one, which is Mistral with a Norm F1 of 0.520.

Family	FOEF F1	Exact F1	Norm F1
Deepseek-LLM	0.458	0.185	0.234
Gemma3	0.697	0.236	0.392
Llama3	0.805	0.119	0.343
Medgemma	0.569	0.134	0.420
Mistral	0.691	0.299	0.520
Qwen3	0.877	0.398	0.689

Table 5.19: FOEF, Exact and Normalized F1 scores of *small* models tested on the extraction task

In the **Medium** category (Table 5.20), Gemma3 and Qwen3 perform similarly on Normalized metrics, though Qwen3 retains superior field detection (FOEF).

Family	FOEF F1	Exact F1	Norm F1
Gemma3	0.778	**0.288**	**0.448**
Qwen3	**0.886**	0.261	0.440

Table 5.20: FOEF, Exact and Normalized F1 scores of *medium* models tested on the extraction task

In the **Large** category (Table 5.21), **MedGemma** reveals to be a very good choice: while it performed poorly in the small version, its large version achieves the second-best Normalized F1 score (0.633), showing a good capability for extracting clinical values when sufficient parameters are available.

Family	FOEF F1	Exact F1	Norm F1
Gemma3	0.861	0.264	0.407
GPT-OSS	0.779	0.222	0.354
Llama3	0.794	0.080	0.328
Medgemma	0.754	**0.476**	**0.633**
Mistral	0.833	0.238	0.396
Qwen3	**0.901**	0.238	0.439

Table 5.21: FOEF, Exact and Normalized F1 scores of *large* models tested on the extraction task

Best Configuration for Extraction

Finally, we compared the Field Description prompt with the more elaborate Complete prompt by considering scores obtained with Large models of **Mistral** and **Gemma 3** families. As seen in Table 5.22, the **Complete** prompt outperformed the **Field Description** one in the Normalized F1 score (0.481 vs 0.435), but not in the FOEF F1 score (0.834 vs 0.841).

Prompt	FOEF F1	Exact F1	Norm F1
Field Description	**0.841**	0.240	0.435
Complete	0.834	**0.320**	**0.481**

Table 5.22: FOEF, Exact and Normalized F1 scores of Field Description and Complete prompts on the extraction task; only scores of Mistral and Gemma 3 *large* models were considered

Applying the repetition technique (Table 5.23) yielded mixed results for the Extraction task. Unexpectedly, for the **Complete** prompt, the single-turn version achieved the highest Normalized F1 (0.491), slightly outperforming both Double (0.469) and Triple (0.483) repetitions. However, for the simpler **Field Description** prompt, repetition consistently improved performance, with the Double version raising the Normalized F1 from 0.401 to 0.453.

Prompt Variation	FOEF F1	Exact F1	Norm F1
Field Description	0.847	0.251	0.401
Double Field Description	**0.888**	**0.291**	**0.453**
Triple Field Description	0.883	0.289	0.443
Complete Prompt	0.899	0.356	**0.491**
Double Complete Prompt	**0.901**	0.341	0.469
Triple Complete Prompt	0.887	**0.389**	0.483

Table 5.23: FOEF, Exact and Normalized F1 scores of Field Description and Complete prompts and respective Repetition variants on the extraction task; only scores of Mistral and Gemma 3 *large* models were considered

We conclude with the response time analysis for these advanced prompts. Figure 5.14 shows how prompt repetition makes the response time longer.

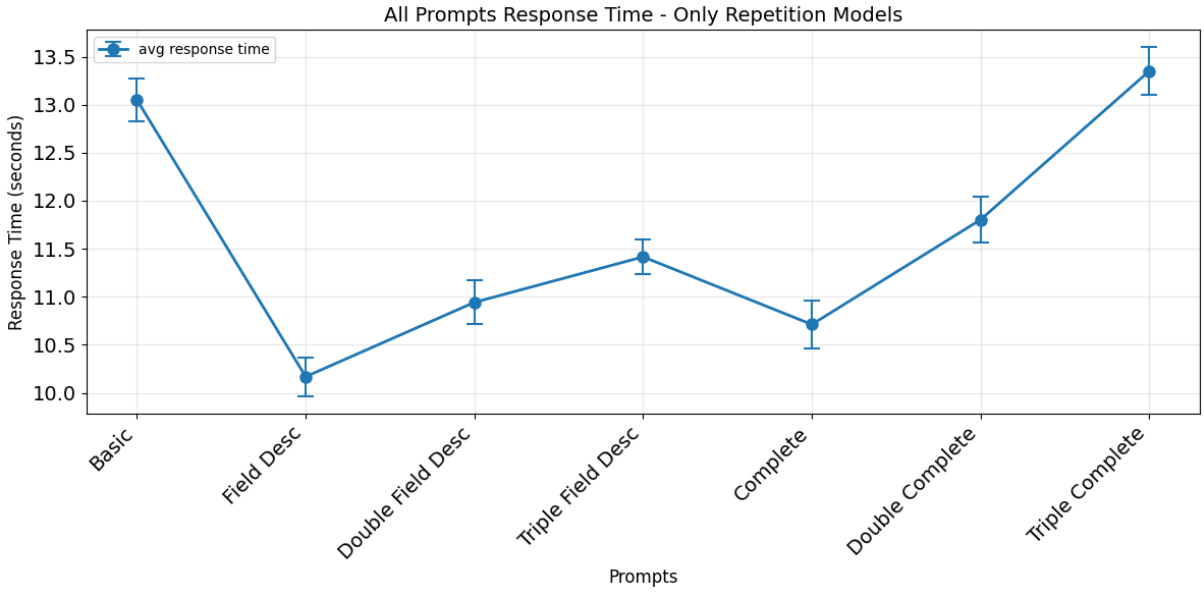


Figure 5.14: Response time and confidence intervals for each extraction task prompt; only times of Mistral and Gemma 3 *large* models were considered

Extraction Task Findings

Unlike for the correction task, increasing model size did not guarantee better performance; in fact, the Qwen3 Small model (roughly 7B parameters) outperformed several Large models. This counter-intuitive result underscores that for structured tasks, the model's instruction-following capability and specific training distribution are more important than raw parameter count. Prompt Repetition confirmed to be an effective prompting technique, allowing to achieve better performance even in this task.

Another factor in common with Correction Task analysis is the stratification of response time. Even though this task required all models to take longer time to answer, Qwen family confirmed to be the slowest. Conversely, Gemma3 and Mistral families still perform similarly to Qwen one, but with significantly lower latency.

5.3. Topic Analysis with BERTopic

As a final exploratory stage of our evaluation, we employed BERTopic [50] to investigate the impact of the correction process on the notes included in the whole dataset.

BERTopic is a modular topic modeling framework that leverages transformers to create clusters of documents allowing for easily interpretable topics. The pipeline typically consists of:

- **Embeddings**: generating document embeddings using a Transformer model. In our implementation, we utilized the *SapBERT* model for the embedding stage, the same used for the evaluation with embedding similarity.
- **Dimensionality Reduction**: reducing dimensionality via UMAP to prepare the data for clustering.
- **Clustering**: using **HDBSCAN** (Hierarchical Density-Based Spatial Clustering of Applications with Noise), a density-based algorithm that identifies clusters of varying shapes and sizes without requiring a pre-specified number of clusters, while also identifying outliers.
- **Topic Representation**: finally extracting topic representations using a **Vectorizer Model** and a class-based **TF-IDF** (c-TF-IDF) procedure. The Vectorizer Model (specifically `CountVectorizer`) transforms documents into a matrix of token counts, while c-TF-IDF calculates the importance of words within a cluster relative to the entire corpus, allowing for the extraction of the most descriptive terms for each topic.

BERTopic on Test Set Notes

First we compared two distinct versions of the test set: the raw original notes and their corresponding "target" (ground truth) version.

The model was initialized with the following configuration:

- **Language**: Italian, to ensure the underlying transformer and tokenizer are optimized for the language of the corpus.
- **Vectorizer Model Configuration**: `CountVectorizer` with an `ngram_range` of $(1, 2)$. This allows the model to capture both individual words (unigrams) and two-word sequences (bigrams).

We chose not to pre-define the number of topics, allowing the algorithm to automatically

determine the most natural thematic structure based on the data's density. All other parameters were kept at their default values.

One particularly interesting observation concerns the outliers, represented by the -1 cluster in BERTopic (documents that do not fit clearly into any identified topic). When transitioning from the raw original notes to the ground truth corrected versions, the number of notes assigned to this noise cluster decreased from 20 to 7, as shown in Table 5.24. This reduction suggests that the correction process successfully standardized the text, resolving ambiguities and shorthand that otherwise hindered the thematic categorization of those documents.

Topic ID	Original Count	Ground Truth Count
-1	20	7
0	43	63
1	25	30
2	12	-

Table 5.24: BERTopic Clustering Results Comparison for the test set.

BERTopic on All the available Notes

Following the test set comparison, we extended this analysis to the entire subset of Federico II dataset available for this thesis. Using the **Mistral Large** model with the **Positive Common Acronyms** prompt, we corrected all the notes and compared the clusters generated by BERTopic with the original and the corrected notes. We initialized the model to force exactly 10 topics for both the original and corrected note sets. While the resulting clusters are not directly comparable due to the stochastic nature of the dimensionality reduction and the shifts in semantic density, this approach proved potentially useful for diagnostic purposes.

The counts obtained for each topic are summarized as follows:

Topic	Original Count	Corrected Count
-1	2070	1836
0	2949	3198
1	722	776
2	636	544
3	536	344
4	506	341
5	334	332
6	54	282
7	29	169
8	26	33
9	-	31

Table 5.25: Comparison of Original and Corrected topic counts for the full dataset

The results for the full dataset seem to confirm the trend observed during the preliminary test set analysis: the "noise" cluster (Topic -1) shows a reduction, decreasing from 2,070 notes in the original set to 1,836 in the corrected version. This shift of documents from the outlier category to defined clusters suggests that the correction process effectively resolved some semantic ambiguities, standardized inconsistent abbreviations, and expanded cryptic shorthand into a more canonical Italian form.

BERTopic provides visualization utilities; notably, it enables the projection of document embeddings into a two-dimensional space using dimensionality reduction techniques like UMAP: in Figures 5.15 and 5.16 we report the output of such method when considering the original notes and their corrected version, respectively.

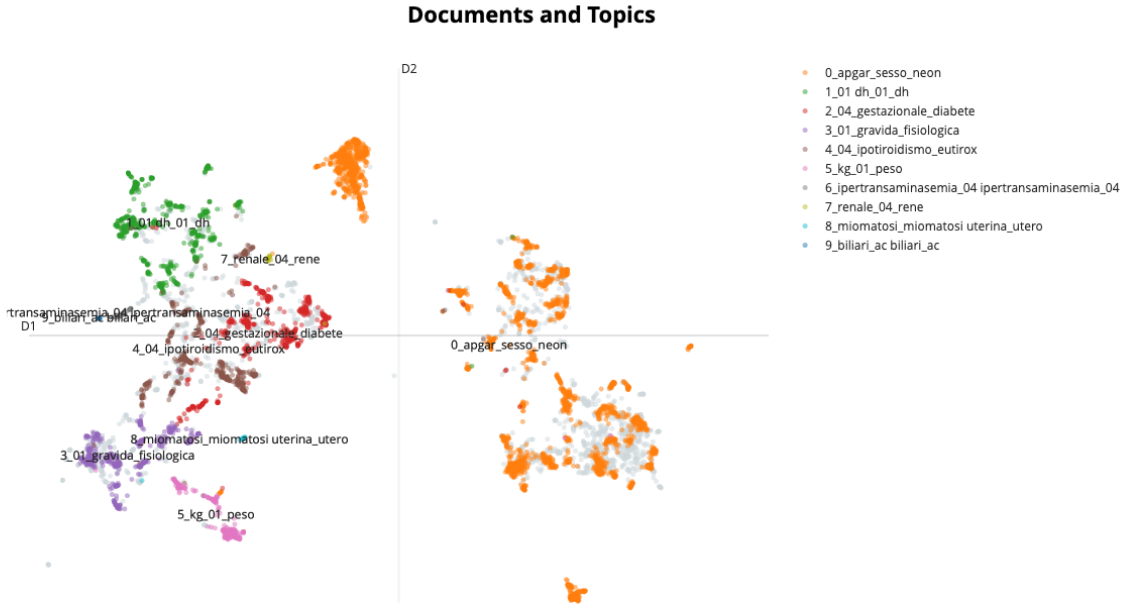


Figure 5.15: BERTopic visualization of the original notes

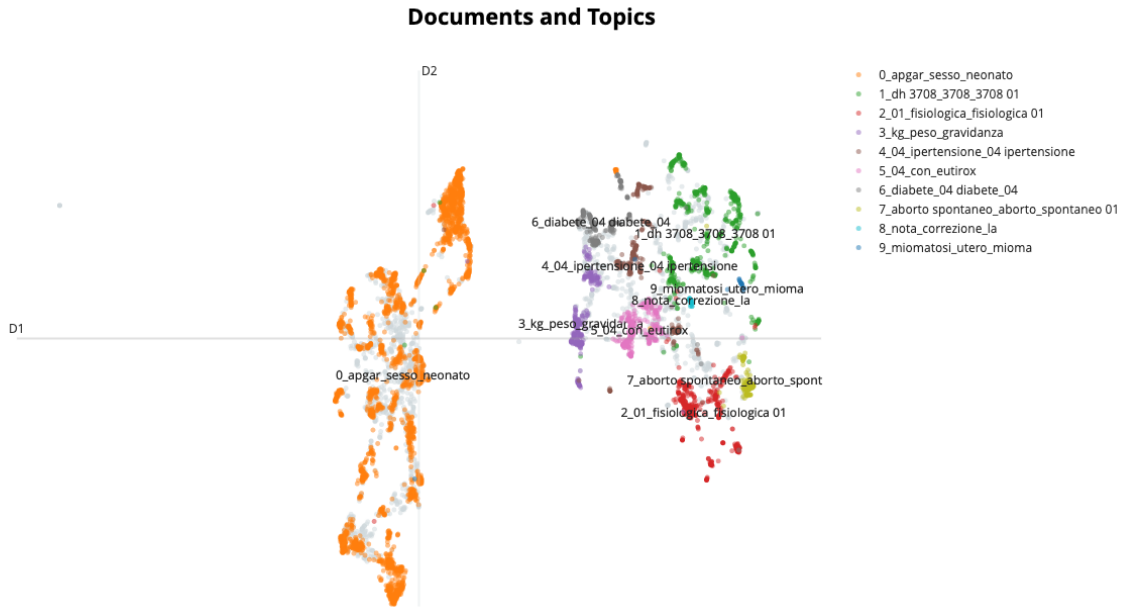


Figure 5.16: BERTopic visualization of the corrected notes

The reason for the asymmetry in the corrected image is due to this small group of notes which have been mostly classfied as -1 cluster by BERTopic but share the fact of citing a "unità esterna" (external unit) in the text.

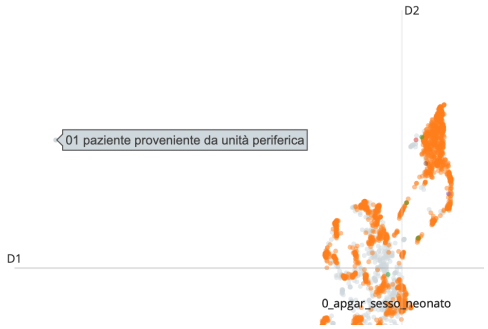


Figure 5.17: BERTopic visualization of the cluster containing the original notes citing "unità esterna". Only one corrected note is shown as example

In the analysis of the corrected notes, a specific cluster emerged: it is Topic 8 and it contains model self-explanations (e.g., *"the note to correct is..."* or *"the correct version of the note is..."*). The presence of this distinct cluster highlights the potential of BERTopic as a "sanity check" for the pipeline: it allows to quickly isolate the failure cases where the model included explanations or answered with the prompt instructions instead of returning only the cleaned clinical text.

Figure 5.18: The most representative words and bigrams for each Topic generated by BERTopic starting from the corrected notes

It is important to emphasize that this approach is intended as a qualitative exploratory tool rather than an evaluation method. These results serve to demonstrate the tool's potential for debugging and pipeline verification. While a deeper collaboration with clinical experts would be required to assign precise medical significance to each cluster, our objective remains to show how this instrument can visualize the transition from raw clinical shorthand to structured, semantically coherent data.

6 | Conclusion and Future Work

This chapter synthesizes the findings from this thesis work, draws conclusions about the viability and impact of using small LLMs for obstetric note processing, and identifies directions for future research. It addresses the research questions posed in the introduction and evaluates the extent to which the identified requirements were met.

6.1. Summary

Challenge Addressed

The primary challenge addressed in this work is the presence of high linguistic noise and lack of structure in Italian clinical obstetric records. These "dark data" repositories are currently unusable for large-scale medical research due to the widespread use of non-standard acronyms, shorthand, and unstructured formatting. Furthermore, the sensitive nature of patient data and the typical resource constraints of hospital IT infrastructures preclude the use of cloud-based APIs, necessitating the development of localized, privacy-preserving solutions.

Methodology

To address these issues, a comprehensive experimental framework was implemented to evaluate the performance of local, open-source Large Language Models (LLMs). The methodology involved testing seven distinct model families, with parameters ranging from 4 to 70 billions, across two specialized tasks: text correction (normalization) and structured data extraction (JSON mapping). The study employed various prompt engineering strategies, specifically focusing on knowledge injection, to bridge the gap between general-purpose models and domain-specific clinical requirements.

Key Findings

The multi-dimensional analysis of the results revealed that "knowledge-augmented" prompts (which provide a simple glossary of obstetric acronyms or a description of the most interesting information to extract), improve performances more than simply increasing the model's parameter count. This underscores the efficiency of targeted prompt engineering in low-resource environments. Additionally, the study highlighted that specialized smaller architectures (e.g., Qwen3 7B) can outperform larger models in strict instruction-following tasks, like information extraction. Finally, an inherent trade-off between inference latency and extraction precision was quantified, identifying the Gemma3 and Mistral families as the most balanced candidates for practical clinical deployment.

In conclusion, this work suggests that locally deployable LLMs are a viable engineering solution for the free-text data cleaning in medical sector. By balancing prompt engineering with careful model selection, it is possible to transform noisy clinical records into a structured asset for the advancement of obstetric research.

6.2. Outputs and Contributions

Concrete Outputs This thesis produced:

A validated experimental pipeline for the evaluation of local LLMs on noisy clinical text.

A ground truth dataset of Italian obstetric notes, cross-referenced and validated by medical experts.

A set of optimized prompt templates specifically designed for clinical text correction and JSON-structured data extraction.

Research Contributions This thesis contributes to the field of Clinical NLP by demonstrating that the correction and information extraction from highly specialized medical text is feasible on consumer-grade hardware. It also shows that architectural optimization and contextual knowledge injection is more important than the raw computational scale. Finally, it provides a pipeline that serves as a ready-to-use framework, offering a solid foundation for further refinements and future research in this domain.

6.3. Limitations

While the pipeline demonstrates promising potential, its current performance remains insufficient for deployment in real-world scenarios.

The performance metrics were calculated against a gold standard comprising only 100 annotated notes, which may not fully represent the complexity of the whole dataset.

Furthermore, the system currently utilizes base model architectures without model-specific fine-tuning and with no systematic hyperparameter optimization.

6.4. Future Work

Future Improvement of the Pipeline

Model-Specific Prompt Tuning In this study, prompt variants were evaluated across different model families using a standardized approach. Future work should investigate model-specific Fine-Tuning to maximize each model's performance.

Retrieval-Augmented Generation (RAG) While the Common Acronyms prompts proved that glossary injection significantly boosts performance, a more scalable evolution would involve RAG. By dynamically retrieving acronym definitions or clinical guidelines from authoritative medical databases, the system could move beyond hard-coded lists and adapt to evolving medical terminology.

Agentic Workflows and Self-Correction Future versions could implement multi-agent loops. A "self-check" mechanism would force the model to parse and validate its own output, correcting hallucinations and inaccuracies before the final answer delivery.

Future Clinical Applications

By transforming "invisible" textual data into standardized, machine-readable information, this NLP pipeline enables downstream clinical applications:

Quantitative Feature Clustering Utilizing the extracted clinical parameters (e.g., Apgar scores, neonatal weight, maternal hypertension) allows for the application of machine learning algorithms to identify patient subgroups without the need for prior labels. Data-driven clusters can subsequently be analyzed to reveal hidden correlations between specific clinical observations and pregnancy complications. For example, such clusters could be used as labels in future research focused on physiological signals like Fetal Heart Rate.

Semantic Embedding Analysis This thesis demonstrated that corrected notes generate embedding vectors (using specialized models like SapBERT) that are semantically closer to the ground truth. These dense vectors can be used to cluster pregnancies based on the entire textual clinical history, capturing information that tabular data alone might miss.

Development of a Clinical Vector Database Integrating these medical embeddings into a Vector Database would represent a significant advancement for hospital research infrastructure. Such a system would enable clinicians to perform "semantic similarity searches" across historical records (e.g., “Find past cases clinically similar to this patient with fetal growth anomalies”), providing a powerful new tool for clinical decision and research.

6.5. Final Remarks

The results of this thesis indicate that locally deployable, open-source Large Language Models represent a promising solution for the "dark data" problem within the medical sector. Even though the constraints of data privacy and limited computational resources, this work demonstrates that the transition from noisy, unstructured obstetric shorthand to standardized digital assets is possible. The evidence suggests that the calibration of prompt engineering, specifically through domain-specific knowledge injection, and the selection of optimized architectures like Gemma3 and Mistral models is often preferable to simply increasing raw parameter counts. This shifts the focus from "larger is better" to "smarter is more sustainable" in clinical NLP applications.

Bibliography

[1] J. H. Holmes, J. Beinlich, M. R. Boland, K. H. Bowles, Y. Chen, T. S. Cook, G. Demiris, M. Draugelis, L. Fluharty, P. E. Gabriel, R. Grundmeier, C. W. Hanson, D. S. Herman, B. E. Himes, R. A. Hubbard, C. E. Kahn Jr., D. Kim, R. Koppel, Q. Long, N. Mirkovic, J. S. Morris, D. L. Mowery, M. D. Ritchie, R. Urbanowicz, and J. H. Moore, "Why Is the Electronic Health Record So Challenging for Research and Clinical Care?," *Methods of Information in Medicine*, vol. 60, no. 1-02, pp. 32–48, 2021.

[2] D. B. Hier, A. S. Kabasakalian, J. S. Brorson, S. S. Alvi, O. Al-Saber, E. Beheshti, S. A. Ezzati, and S. T. Carmichael, "Preprocessing of Physician Notes by LLMs Improves Clinical Concept Extraction Without Information Loss," *Information*, vol. 16, p. 42, 2025.

[3] J. A. Smit, R. Van der Graaf, M. Mostert, I. Vaartjes, M. Zuidgeest, D. Grobbee, and J. J. M. van Delden, "Overcoming ethical and legal obstacles to data linkage in health research: Stakeholder perspectives," *International Journal of Population Data Science*, vol. 8, no. 1, 2023.

[4] L. Hays, J. Weaver, M. Gauger, N. Buckner, B. Bailey, A. Stone, and L. Orlando, "Barriers to leveraging valuable health data for collaborative patient care: How will we integrate family health histories?," *Systems*, vol. 13, no. 3, p. 140, 2025.

[5] S. M. Williamson and V. Prybutok, "Balancing privacy and progress: A review of privacy challenges, systemic oversight, and patient perceptions in AI-driven healthcare," *Applied Sciences*, vol. 14, no. 2, p. 675, 2024.

[6] S. Calcagno, A. Calvagna, E. Tramontana, and G. Verga, "Merging ontologies and data from Electronic Health Records," *Future Internet*, vol. 16, no. 2, p. 62, 2024.

[7] E. Liscio, G. Campagna, *et al.*, "Advancing Clinical Data Standardization: LLM-based Mapping of Italian Hospital Notes to International Ontologies," in *Proceedings of the 23rd International Conference on Artificial Intelligence in Medicine (AIME 2025), Pavia, Italy*, Springer, 2025.

[8] M. Moor, O. Banerjee, Z. S. Hossein Abad, H. M. Krumholz, J. Leskovec, E. J. Topol, and P. Rajpurkar, "Foundation models for generalist medical Artificial Intelligence," *Nature*, vol. 616, pp. 259–265, 2023.

[9] L. Lilli, C. Masciocchi, A. Marchetti, G. Arcuri, and S. Patarnello, "Prompting Large Language Models for Italian Clinical Reports: A Benchmark Study," in *BioNLP 2025*, pp. 190–200, 2025.

[10] L. Builtjes, B.-J. G. L. M. Groeneveld, R. Grundmeier, M. D. Ritchie, and D. Kim, "Leveraging open-source Large Language Models for clinical information extraction in resource-constrained settings," *JAMIA Open*, vol. 8, no. 5, p. ooaf109, 2025.

[11] E. Spairani, B. Daniele, G. Magenes, and M. G. Signorini, "A Novel Large Structured Cardiotocographic Database," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3965–3968, IEEE, 2022.

[12] M. T. Chiaravalloti, G. Serratore, F. Del Ben, and A. Steffan, "LOINC mapping experiences in Italy: The case of Friuli-Venezia Giulia region," in *Proceedings of the 18th International Joint Conference on Biomedical Engineering Systems and Technologies*, pp. 959–967, 2025.

[13] L. Wang, W. Bi, S. Zhao, Y. Ma, L. Lv, C. Meng, J. Fu, and H. Lv, "Investigating the Impact of Prompt Engineering on the Performance of Large Language Models for Standardizing Obstetric Diagnosis Text: Comparative Study," *JMIR Formative Research*, vol. 8, p. e51234, 2024.

[14] E. G. Woo, I. Zighelboim, T. Gifford, J. G. Bell, H. Milthorpe, E. Alsentzer, R. E. Longman, J. E. Tolosa, and B. K. Beaulieu-Jones, "Predicting Postpartum Hemorrhage Using Clinical Features Extracted With Large Language Models," *Obstetrics & Gynecology*, vol. 2, no. 5, 2025.

[15] Ollama, "Ollama website." `https://ollama.com/`, accessed 2025-12-01.

[16] Ollama, "Ollama github." `https://github.com/ollama/ollama`, accessed 2025-07-01.

[17] G. Varisco, G. Steyde, E. Peri, I. Hoogendoorn, M. G. Signorini, J. O. E. H. van Laar, M. Mischi, and M. B. van der Hout-van der Jagt, "Predicting Intrapartum Acidemia: A Review of Approaches Based on Fetal Heart Rate," *Bioengineering*, vol. 13, no. 2, p. 146, 2026.

[18] L. Mendis, M. Palaniswami, F. Brownfoot, and E. Keenan, "Computerised Car-

diotocography Analysis for the Automated Detection of Fetal Compromise during Labour: A Review," *Bioengineering*, vol. 10, no. 10, p. 1007, 2023.

[19] R. M. Grivell, Z. Alfirevic, G. M. Gyte, and D. Devane, "Antenatal cardiotocography for fetal assessment," *Cochrane Database of Systematic Reviews*, no. 9, p. CD007863, 2015.

[20] E. Spairani, G. Steyde, F. Spuri Forotti, G. Magenes, and M. G. Signorini, "Prediction of IUGR condition at birth by means of CTG recordings and a ResNet model," *Computers in Biology and Medicine*, vol. 190, p. 110123, 2025.

[21] M. G. Signorini, N. Pini, A. Malovini, R. Bellazzi, and G. Magenes, "Integrating machine learning techniques and physiology based Heart Rate features for antepartum fetal monitoring," *Computer Methods and Programs in Biomedicine*, vol. 185, p. 105015, 2020.

[22] M. G. Signorini, A. Fanelli, and G. Magenes, "Monitoring Fetal Heart Rate during Pregnancy: Contributions from Advanced Signal Processing and Wearable Technology," *Computational and Mathematical Methods in Medicine*, vol. 2014, p. 707581, 2014.

[23] L. Wang, Y. Ma, W. Bi, H. Lv, and Y. Li, "An Entity Extraction Pipeline for Medical Text Records Using Large Language Models: Analytical Study," *Journal of Medical Internet Research*, vol. 26, p. e54580, 2024.

[24] H. Adam, J. Lin, J. Lin, H. Keenan, A. Wilson, and M. Ghassemi, "Clinical Information Extraction with Large Language Models: A Case Study on Organ Procurement," 2025.

[25] C. Martinelli, A. Giordano, V. Carnevale, S. R. Burk, L. Porto, G. Vizzielli, and A. Ercoli, "The PERFORM Study: Artificial Intelligence Versus Human Residents in Cross-Sectional Obstetrics-Gynecology Scenarios Across Languages and Time Constraints," *Mayo Clinic Proceedings: Digital Health*, vol. 3, no. 2, p. 100206, 2025.

[26] American College of Obstetricians and Gynecologists, "Committee Opinion No. 644: The Apgar Score," *Obstetrics & Gynecology*, vol. 126, no. 4, pp. e52–e55, 2015.

[27] J. Yuan, H. Li, X. Ding, W. Xie, Y.-J. Li, W. Zhao, K. Wan, J. Shi, X. Hu, and Z. Liu, "Understanding and Mitigating Numerical Sources of Nondeterminism in LLM Inference," *arXiv preprint arXiv:2506.09501*, 2025.

[28] Ollama, "Ollama library." `https://ollama.com/library`, accessed 2025-12-01.

[29] Ollama, "Gemma 3 models page." `https://ollama.com/library/gemma3`, accessed 2025-10-01.

[30] Ollama, "Medgemma models page." `https://ollama.com/alibayram/medgemma`, accessed 2025-10-01.

[31] Ollama, "Llama 3 models page." `https://ollama.com/library/llama3`, accessed 2025-10-01.

[32] Ollama, "Mistral models page." `https://ollama.com/library/mistral`, accessed 2025-10-01.

[33] Ollama, "Mistral models page." `https://ollama.com/library/mistral-small3.2`, accessed 2025-10-01.

[34] Ollama, "Deepseek models page." `https://ollama.com/library/deepseek-llm:7b-chat-q3_K_M`, accessed 2025-10-01.

[35] Ollama, "Qwen 3 models page." `https://ollama.com/library/qwen3`, accessed 2025-10-01.

[36] Ollama, "Gpt-oss models page." `https://ollama.com/library/gpt-oss`, accessed 2025-10-01.

[37] L. Boonstra, *Prompt Engineering*. Mountain View, CA: Google, September 2024.

[38] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *CoRR*, vol. abs/2005.11401, 2020.

[39] Z. Chen, Y. Liu, L. Shi, X. Chen, Y. Zhao, and F. Ren, "MDEval: Evaluating and Enhancing Markdown Awareness in Large Language Models," 2025.

[40] P. Sharma and A. Z. Henley, "Modular Prompt Optimization: Optimizing Structured Prompts with Section-Local Textual Gradients," 2026.

[41] Y. Leviathan, M. Kalman, and Y. Matias, "Prompt Repetition Improves Non-Reasoning LLMs," 2025.

[42] Ollama, "Ollama: Generating Structured Output JSON with a Schema." Accessed 2025-11-01.

[43] Pydantic, "Pydantic: BaseModel.model_json_schema." Accessed 2025-07-01.

[44] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," 2020.

[45] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," pp. 311–318, July 2002.

[46] Z. R. K. Rostam, M. Takács, and G. Kertész, "Evaluating Large Language Models: A Review of Metrics and Benchmarks," in *Proceedings of the 2025 IEEE 23rd Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 233–240, 2025.

[47] CambridgeLTL, "SapBERT-UMLS-2020AB: All-lang from XLM-R-large." Accessed 2025-10-01.

[48] Intfloat, "Multilingual E5 Large Text Embeddings." Accessed 2025-10-01.

[49] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Multilingual E5 Text Embeddings: A Technical Report," 2024.

[50] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," *arXiv preprint arXiv:2203.05794*, 2022.

A | Appendix

A.1. Common Acronyms

Acronym	Italian Expansion	English Translation
p.s.	parto spontaneo	spontaneous delivery
t.c.	taglio cesareo	cesarean section
mef	morte endouterina fetale	fetal intrauterine death
ivg	interruzione volontaria di gravidanza	voluntary termination of pregnancy
ab/abs	aborto spontaneo	spontaneous abortion
+ivg/+ab	pregresso	previous
gr	gravidanza	pregnancy
pa	pressione arteriosa	arterial pressure
cc	circonferenza cranica	head circumference
n/neon	neonato	newborn
tin	terapia intensiva neonatale	neonatal intensive care
tia	attacco ischemico transitorio	transient ischemic attack
geu	gravidanza extrauterina	ectopic pregnancy
mipp	minaccia di parto pretermine	threat of preterm delivery
p.o.v.e.	parto operativo con ventosa	operative delivery with vacuum extraction
icsi	iniezione intracitoplasmatica dello spermatozoo	intracytoplasmic sperm injection
ass. di mano	associazione di mano	hand association

Table A.1: Common acronyms found in obstetric notes

A.2. Clinically Relevant Values

This appendix provides a list of all clinically relevant values in the obstetric notes, using in two tables: one with the original Italian names and descriptions, and one with their English translations.

Original Italian Field Names and Descriptions

Field Name	Description
codice ambulatoriale	Codice ambulatoriale del paziente, se presente si trova all'inizio della nota, è un numero a due cifre tra 01 e 12
reparto	Reparto in cui il paziente è ricoverato, solitamente descritto da un numero di 4 cifre
dottore	Dottore respomsabile della visita
tipologia parto	Tipologia di parto (sponaneo (PS), cesareo(TC), urgente)
data nascita	Data di nascita del neonato, scrivila come gg/mm/aaaa
pH	Potenziale idrogeno
pCO2 mmHg	Pressione parziale di CO2 in mmHg
pO2 mmHg	Pressione parziale di O2 in mmHg
HCO3 act mEq L	Bicarbonato attivo in mEq/L
HCO3 std mEq L	Bicarbonato standard in mEq/L
Base excess in blood mmol L	Base excess nel sangue in mmol/L
Base excess in ecf mmol L	Base excess nel fluido extracellulare in mmol/L
Base excess mmol L	Base excess totale in mmol/L
ctCO2 mmol L	CO2 totale in mmol/L
lattato mmol L	Lattato, spesso indicato da LAT o LAC
peso neonato	Peso del paziente in kg
	continues in next page

Table A.2: Clinically relevant values: Original Italian field names and descriptions

Field Name	Description
lunghezza neonato	Lunghezza del paziente in cm
circonferenza cranica	Circonferenza della testa in cm
apgar1	Primo valore apgar
apgar2	Secondo valore apgar
apgar	Valori apgar separati da /, ad esempio 8/9
sesso	Sesso del neonato: M se maschio, F se femmina, campo vuoto se l'informazione è assente
peso materno	Peso della madre in kg
guadagno peso materno	Guadagno di peso della madre in gravidanza in kg, spesso indicato come +X kg o -X kg
ipertensione	Presenza di iperensione: sì/no; specificare eventuali dettagli, esempio 'gestazionale'
diabete	Presenza di diabete: sì/no; specificare eventuali dettagli, esempio 'gestazionale' o il tipo
obesità	Presenza di obesità: sì/no; specificare eventuali dettagli, esempio 'gestazionale'
extra	Informazioni della nota non riportate nei campi precedenti

Table A.3: Clinically relevant values: continuation

English Translation of Field Names and Descriptions

Field Name	Description
outpatient code	Patient's outpatient code; if present, it is found at the beginning of the note and consists of a two-digit number between 01 and 12
ward	The ward where the patient is admitted, usually described by a 4-digit number
doctor	Doctor responsible for the visit
delivery type	Type of delivery (spontaneous (PS), cesarean (TC), urgent)
birth date	Newborn's date of birth, written as dd/mm/yyyy
pH	Potential of hydrogen
pCO2 mmHg	Partial pressure of CO2 in mmHg
pO2 mmHg	Partial pressure of O2 in mmHg
HCO3 act mEq L	Active bicarbonate in mEq/L
HCO3 std mEq L	Standard bicarbonate in mEq/L
Base excess in blood mmol L	Base excess in blood in mmol/L
Base excess in ecf mmol L	Base excess in extracellular fluid in mmol/L
Base excess mmol L	Total base excess in mmol/L
ctCO2 mmol L	Total CO2 in mmol/L
lactate mmol L	Lactate, often indicated by LAT or LAC
newborn weight	Weight of the patient in kg
	continues in next page

Table A.4: Clinically relevant values: English field names and descriptions

Field Name	Description
newborn length	Length of the patient in cm
head circumference	Circumference of the head in cm
apgar1	First Apgar score
apgar2	Second Apgar score
apgar	Apgar values separated by /, for example 8/9
sex	Sex of the newborn: M for male, F for female; empty field if the information is missing
maternal weight	Weight of the mother in kg
maternal weight gain	Mother's weight gain during pregnancy in kg, often indicated as +X kg or -X kg
hypertension	Presence of hypertension: yes/no; specify details, e.g., 'gestational'
diabetes	Presence of diabetes: yes/no; specify details, e.g., 'gestational' or type
obesity	Presence of obesity: yes/no; specify details, e.g., 'gestational'
extra	Information from the note not reported in the previous fields

Table A.5: Clinically relevant values: continuation

A.3. Full Complete Prompts

A.3.1. Correction Complete Prompt

```
‘‘# RUOLO
Agisci come un’intelligenza artificiale specializzata nella
revisione e correzione di note manuali ostetriche. Il tuo compito
è normalizzare il testo per renderlo leggibile e professionale.

# CONTESTO E GLOSSARIO
Utilizza il seguente glossario per l’espansione degli acronimi,
prestando attenzione al contesto clinico:
{Comprehensive glossary with 12+ acronyms, each with contextual
notes}

# ISTRUZIONI OPERATIVE
1. **Espansione e Correzione**: Espandi le abbreviazioni secondo il
glossario e correggi eventuali errori di battitura.
2. **Conservazione**: Mantieni tutte le informazioni della nota
originale, espandendole dove necessario per chiarezza.
3. **Gestione Ambiguità**: Se un acronimo non è presente nel
glossario o risulta ambiguo, mantienilo inalterato.
4. **Integrità**: Non aggiungere diagnosi o dati non presenti nel
testo originale.
5. **Formato Output**: Rispondi esclusivamente con la nota
corretta. Ometti spiegazioni, commenti o saluti.

continues in next page
```

```
...
# ESEMPI (FEW-SHOT)
Input:  "Donna 70 kg - 12 kg, p.s., neon 3200g M."
Output:  "Donna peso materno 70 kg aumentato in gravidanza di 12 kg,
parto spontaneo, neonato 3200g maschio."

Input:  "Pregressa geu e +ab."
Output:  "Pregressa gravidanza extrauterina e pregresso aborto
spontaneo."

Input:  "Ricovero per mipp a 32 EG."
Output:  "Ricovero per minaccia di parto pretermine a 32 settimane di
età gestazionale."

# NOTA DA CORREGGERE:
{Nota}
```

Correction Complete Prompt - English

```
‘‘# ROLE
Act as an artificial intelligence specialized in the review and
correction of manual obstetric notes.  Your task is to normalize
the text to make it readable and professional.

# CONTEXT AND GLOSSARY
Use the following glossary for the expansion of acronyms, paying
attention to the clinical context:
{Comprehensive glossary with 12+ acronyms, each with contextual
notes}

# OPERATING INSTRUCTIONS
1.  **Expansion and Correction**:  Expand abbreviations according to
the glossary and correct any typos.
2.  **Preservation**:  Maintain all information from the original
note, expanding it where necessary for clarity.
3.  **Ambiguity Management**:  If an acronym is not present in the
glossary or is ambiguous, keep it unchanged.
4.  **Integrity**:  Do not add diagnoses or data not present in the
original text.
5.  **Output Format**:  Respond exclusively with the corrected note.
Omit explanations, comments, or greetings.
```

continues in next page

```
...
# EXAMPLES (FEW-SHOT)
Input:  "Woman 70 kg - 12 kg, s.d., newb 3200g M."
Output:  "Woman maternal weight 70 kg increased during pregnancy by
12 kg, spontaneous delivery, newborn 3200g male."

Input:  "Previous geu e +ab."
Output:  "Previous ectopic pregnancy and previous spontaneous
abortion."

Input:  "Hospitalization for mipp at 32 GE."
Output:  "Hospitalization for threat of preterm labor at 32 weeks of
gestational age."

# NOTE TO BE CORRECTED:
{Note}
```

A.3.2. Extraction Complete Prompt

```
# RUOLO
Agisci come un esperto analista di dati clinici neonatali. Il
tuo compito è estrarre informazioni strutturate da note ostetriche
italiane.

# ISTRUZIONI DI ESTRAZIONE
1. **Formato di Output**: Restituisci esclusivamente un oggetto
JSON che segua rigorosamente lo schema fornito.
2. **Dati Mancanti**: Se un valore non è presente nella nota,
inserisci una stringa vuota (“”).
3. **Precisione**: Estrai i valori numerici mantenendo le unità di
misura indicate nelle descrizioni.
4. **Sesso**: Usa esclusivamente “M”, “F” o stringa vuota. 5.
**Apgar**: Estrai i valori individuali (apgar1, apgar2) e la coppia
formattata (apgar).
6. **Ipertensione/Diabete/Obesità**: Rispondi “sì” o “no”. Se
presente, aggiungi il dettaglio (es: “sì, gestazionale”).

# SCHEMA JSON (Campi da estrarre)
{get_fields_descriptions}

here we recall Get Fields Descriptions pseudo code,
prompt continues in next page
```

Algorithm A.1 Get Fields Descriptions

```
descriptions = ""
for field in fields do
    desc = field[description]
    descriptions.append(field + ": " + desc)
    descriptions.append(newline)
end for
return descriptions
```

```
...
# ESEMPIO (FEW-SHOT)
Input: ‘‘03. Reparto 4012. Dr. Rossi. Parto PS a 39+0. Nato
M, peso 3.250kg, lungo 50cm, CC 34. Apgar 8/9. pH 7.25, LAT 2.1.
Madre peso 70kg (+12kg).’’

Output:
{
"codice_ambulatoriale": "03",
"reparto": "4012",
"dottore": "Rossi",
"tipologia_parto": "spontaneo (PS)",
...
"pH": "7.25",
"lattato_mmol_L": "2.1",
"peso_neonato": "3.250",
...
"apgar": "8/9",
"sesso": "M",
"peso_materno": "70",
"guadagno_peso_materno": "12",
"ipertensione": "no",
"diabete": "no",
"obesità": "no",
"extra": "39+0 EG"
}

# NOTA CLINICA DA ELABORARE:
{nota}
```

Extraction Complete Prompt - English

```
# ROLE
Act as an expert neonatal clinical data analyst. Your task is to
extract structured information from Italian obstetric notes.

# EXTRACTION INSTRUCTIONS
1. **Output Format**: Return exclusively a JSON object that
strictly follows the provided schema.
2. **Missing Data**: If a value is not present in the note, insert
an empty string (“”).
3. **Precision**: Extract numerical values while maintaining the
units of measurement indicated in the descriptions.
4. **Sex**: Use exclusively “M”, “F” or an empty string.
5. **Apgar**: Extract individual values (apgar1, apgar2) and the
formatted pair (apgar).
6. **Hypertension/Diabetes/Obesity**: Answer “yes” or “no”. If
present, add the detail (e.g.: “yes, gestational”).

# JSON SCHEMA (Fields to extract)
{get_fields_descriptions}

here we recall Get Fields Descriptions pseudo code,
prompt continues in next page
```

Algorithm A.2 Get Fields Descriptions

```
descriptions = ""
for field in fields do
    desc = field[description]
    descriptions.append(field + ": " + desc)
    descriptions.append(newline)
end for
return descriptions
```

```
...
# EXAMPLE (FEW-SHOT)
Input: ‘‘03. Ward 4012. Dr. Rossi. Delivery PS at 39+0. Born M,
weight 3.250kg, length 50cm, CC 34. Apgar 8/9. pH 7.25, LAT 2.1.
Mother weight 70kg (+12kg).’’

Output:
{
"clinical_code": "03",
"ward": "4012",
"doctor": "Rossi",
"delivery_type": "spontaneous (PS)",
...
"pH": "7.25",
"lactate_mmol_L": "2.1",
"neonatal_weight": "3.250",
...
"apgar": "8/9",
"sex": "M",
"maternal_weight": "70",
"maternal_weight_gain": "12",
"hypertension": "no",
"diabetes": "no",
"obesity": "no",
"extra": "39+0 GA"
}

# CLINICAL NOTE TO PROCESS:
{note}
```

List of Figures

List of Tables

Acknowledgements

I would like to thank Prof. Maria Gabriella Signorini for the opportunity of working on this thesis and the trust she gave me since the first time we talked about it.

I also wish to thank Giulio Steyde and Pierluigi Reali for guiding me through the long and complex path of this project, and Prof. Mark J. Carman for his advice regarding Large Language Models and their occasionally incomprehensible nature.

Finally, my thanks go to Giulia Belliero, the *obstetrician* cited throughout this thesis that helped me to understand the meaning of the notes.