GREYC at CRF Filling 2026: Rewrite Before You Extract - Rewriting Clinical Notes for Automated CRF

Jesus Lovon-Melgarejo, Jérémie Pantin, Gaël Dias
Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ,
GREYC UMR6072, F-14000 Caen, France
{jesus.lovon,jeremie.pantin, gael.dias}@unicaen.fr

Abstract

This paper describes the system we submitted to the CRF:filling 2026 shared task. We propose a modular, LLM-based framework including an LLM as rewriter, which enhances the original clinical note from the perspective of each target CRF item; an LLM as an extractor, which retrieves the relevant value using a k-shot prompting strategy; and an LLM as a judge, which determines whether the clinical note contains evidence to support a given answer, defaulting to *unknown* otherwise. We evaluated our system on the English portion of the dataset; our complete framework achieves a macro-F1 of 0.64 on the development set. Our analysis reveals that while the rewriting step effectively generates correct factual information, it also increases false positives. The judge component mitigates this by adopting a conservative prediction strategy that substantially reduces false positives at the cost of a moderate reduction in true positives, yielding higher precision and better alignment with the shared task metric. On the test set, a light version of our system ranked 21 out of 32 public submissions, achieving a macro-F1 of 0.45.

Keywords: rewrite, LLMs, case report form

1. Introduction

A Case Report Form (CRF) is a standardised document widely used in clinical research to collect patient data across diverse studies and healthcare environments (Bellary et al., 2014). Each CRF contains a fixed set of items to be populated with patient medical information. Through their standardised structure, CRFs enable consistent data collection, ensuring accuracy, reliability, and validity required for reproducible clinical findings.

However, CRFs are typically filled manually from clinical notes and electronic medical records, a process that is time-consuming and prone to inconsistencies. Automating the population of CRFs from clinical narratives would therefore be highly beneficial: it could accelerate clinical research, reduce the manual workload of healthcare professionals, and produce structured representations of patient information. Consequently, recent research has explored automated systems that populate CRFs using information extracted from clinical notes and electronic medical records (Mac Kenzie et al., 2016; Gutiérrez-Sacristán et al., 2024).

The CRF filling 2026 shared task (Ferrazzi et al., 2026) aims to advance the development of systems applicable to real-world clinical settings. The datasets comprise CRFs for patients presenting dyspnea, using data from an Emergency Department in Italian and English.

Large Language Models (LLMs) have demonstrated remarkable capabilities in language understanding and generation (Petroni et al., 2019), with strong performance on health-related tasks such as clinical prediction, medical knowledge understanding, and question-answering (Xu et al., 2024; Chen et al., 2023; Wu et al., 2024). Based on previous work that demonstrated that rewriting clinical patient profiles improves downstream performance, whether by generating new knowledge with LLMs (Sui et al., 2024), refining existing information using their internal knowledge (Lovon-Melgarejo et al., 2026), or augmenting profiles with external sources (Xu et al., 2024), we propose a multi-stage LLM framework for automated CRF filling.

In this work, we propose a multi-stage LLM-based framework for automated CRF filling. Our system consists of three components. First, an *LLM as rewriter* reformulates and enriches the original clinical input. Second, an *LLM as an extractor* retrieves the expected CRF item values, following previous approaches for structured information extraction (Ferrazzi et al., 2025). Third, to mitigate hallucination, where an LLM produces an answer despite insufficient supporting evidence (Su et al., 2022; Zhang et al., 2024), an *LLM as a judge* determines whether each extracted item can be reliably inferred from the provided context. Additionally, our approach accurately minimizes false positives and false negatives, preventing misleading entries.

2. Data

The CRF-filling dataset (Kaczmarek et al., 2026) consists of 290 anonymised emergency care clinical notes collected from the San Giovanni Bosco Hospital (Italy), covering patients admitted between January 2021 and December 2023. Each note is paired with a CRF capturing patient information

Proceedings of the Third Workshop on Patient-Oriented Language Processing (CL4Health), pages 384–389
12 May 2026.

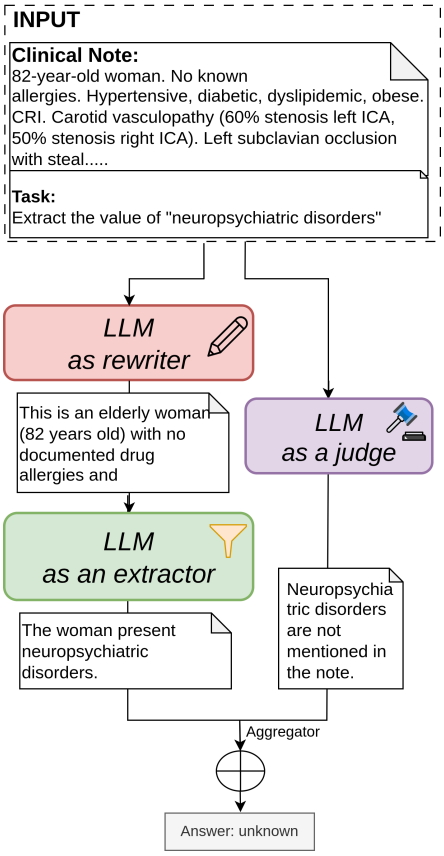

Figure 1: Overview of our framework architecture using three LLM-based components.

across a predefined set of medical items. The task focuses on the *dyspnea CRF*, which contains 134 items. The dataset is provided in both English and Italian and is split into training (10 notes), development (90 notes), and test (200 notes) sets.

Each CRF item is restricted to a predefined set of values depending on the type of information it represents, including binary (yes/no), ordinal (below/within/above), or measured/unknown categories used for tests without fixed categorical outcomes. The *unknown* value is valid for any item, enabling the schema to account for information absent from the clinical note. In addition to the manually annotated data, the task release includes 71 semi-automatically annotated note–CRF pairs in English (70 in Italian) (Ferrazzi et al., 2025), as well as 2667 unannotated clinical notes related to dyspnea, which were not used in our experiments.

3. Methodology

In this section, we describe our system, which comprises three LLM-based components: a rewriter, an information extractor, and a judge. We also consider an aggregator operator (Figure 1). Following previous work (Kaczmarek et al., 2026; Ferrazzi et al., 2025), we adopt a single-item processing strategy to avoid cascading errors, processing each of the 134 CRF items independently, resulting in 134 inference steps per entry in the evaluation set. For each entry, our framework operates through two parallel branches. In the first branch, an *LLM as rewriter* explicitly enriches the clinical note before a second *LLM extracts* the relevant information for the target CRF item. In the second branch, an *LLM as a judge* evaluates whether the original input contains sufficient evidence to answer the item, acting as a safeguard against hallucinations introduced in the first branch.

3.1. LLM as Rewriter

We prompt the LLM to complete the clinical note by interpreting and reorganising its content from the perspective of the target CRF item. This produces an enriched clinical note that either highlights information relevant to the item or explicitly states that no such information is present. Through this rewriting step, the parametric knowledge encoded in the LLM can be transferred and leveraged to enrich the clinical patient profile (Sui et al., 2024). The prompt used for the rewriter is as follows:

LLM prompt

You are an expert clinician specialized in clinical record analysis. Your task is to extract and interpret a specific clinical parameter from a patient's clinical history. Your input is: a patient clinical history and a clinical question. Your output must be a structured clinical summary of the relevant facts from the profile, written so that a reader could easily answer the question.
PATIENT: {patient profile}
QUESTION: What is the link between the patient profile and {CRF item}?

3.2. LLM as an Extractor

Following (Kaczmarek et al., 2026; Ferrazzi et al., 2025), we employ an LLM for information extraction using a k-shot prompting strategy, providing as input the concatenation of the original clinical note and the rewritten profile. The used prompt is:

LLM prompt

You are an expert clinical information extractor. You will be given a patient's clinical history, an interpretation and a question. Your role is to answer the question strictly based on the provided information.
PATIENT: {patient profile}
INTERPRETATION: {rewritten patient profile}
QUESTION: What are the results and measures of {CRF item}?

3.3. LLM as a Judge

LLMs are prone to hallucination, generating plausible-sounding answers even when the clinical note provides no supporting evidence for a given CRF item. To mitigate this, we introduce an *LLM as a judge* tasked with determining whether a given clinical note contains sufficient information to reliably answer a target CRF item. The prompt used for this component is:

LLM prompt

You are an expert clinical information extractor. You will be given a patient's clinical history, an interpretation and a question. Your role is to answer the question strictly based on the provided information.
PATIENT: {patient profile}
INTERPRETATION: {rewritten patient profile}
QUESTION: According to the clinical information provided, is the value of item explicitly mentioned or documented?

Aggregator Finally, the outputs of the *LLM as a judge* and *LLM as an extractor* components are aggregated to produce the final predicted value. Let $J(n, q) \in \{\text{yes}, \text{no}\}$ denote the binary output of the *LLM as a judge*, indicating whether the answer to item q is present in clinical note n. Let $E(n, q)$ denote the value predicted by the *LLM as an extractor* for item q given note n. The aggregator operator $\oplus$ for an item q is defined as:

$$\oplus(n, q) = \begin{cases} E(n, q) & \text{if } J(n, q) = \text{'yes'} \\ \textit{unknown} & \text{otherwise} \end{cases} \quad (1)$$

4. Experimental Setup

We evaluated our approach across multiple open weight LLMs. For general-purpose models, we tested Qwen$_s$[1], Qwen[2], and Llama[3]. We additionally considered one domain-specific model: MedLlama[4](Med42), a medically-oriented LLM.

For the *extractor*, we applied k-shot prompting ($k = 3$), selecting examples matching the target CRF item from the training split, and evaluated all four models. For the *rewriter*, we considered Qwen$_s$, Qwen, and Llama. For *the judge*, we used the Qwen model.

Regarding inference parameters, the *rewriter* and *judge* used a temperature of 0.7, top-p of 0.9, and a maximum of 512 generated tokens. The *extractor* used a temperature of 0 and top-p of 1 to ensure deterministic outputs. For the *judge* specifically, rather than relying on generated output, we recover the logit values of the first valid generated token to determine a binary yes/no decision.

Model	Macro-F1	
	w/o judge	w/ judge
	w/o rewriter	
Med42	0.3441	0.5691
Llama	0.3776	0.5846
Qwen	0.3937	0.5947
Qwen$_s$	0.5046	<u>0.6293</u>
	w/ rewriter	
Llama→Med42	0.3236	0.5339
Qwen→Med42	0.3394	0.5883
Qwen$_s$→Med42	0.3716	0.6011
Qwen$_s$→Llama	0.4275	0.6026
Qwen→Qwen$_s$	0.4496	**0.6445**

Table 1: Macro-F1 scores for the development (dev) set in English. We highlight **best** and <u>second best</u> results. Blue and red cells indicate our baselines and final framework, respectively.

Metrics We evaluate our system using the official task metric, macro-F1. To enable a more in-depth analysis, we additionally report precision, recall, true positives, and false positives.

5. Results

We performed an in-depth evaluation on the development set, where gold annotations are available. Table 1 reports the results across all evaluated configurations. We report only the best configurations in our tables. Our baselines, using a 3-shot setup,

[1] Qwen/Qwen3-8B
[2] Qwen/Qwen3-32B
[3] meta-llama/Meta-Llama-3.1-70B-Instruct
[4] m42-health/Llama3-Med42-8B

Model	w/o judge				w/ judge			
	TP↑	FP↓	P↑	R↑	TP↑(Δ%)	FP↓(Δ%)	P↑(Δ%)	R↑(Δ%)
w/o rewriter								
Med42	264	2061	0.11	0.77	207(-22%)	174(-92%)	0.54(+391%)	0.58(-25%)
Llama	282	2078	0.12	**0.87**	231(-18%)	248(-88%)	0.48(+300%)	0.63(-28%)
Qwen	300	1522	0.16	0.85	211(-30%)	**103**(-93%)	**0.67**(+319%)	0.56(-34%)
Qwen$_s$	243	**349**	**0.41**	0.65	208(-14%)	107(-69%)	0.66(+61%)	0.55(-15%)
w/ rewriter								
Llama→Med42	277	1910	0.13	0.83	203(-27%)	177(-91%)	0.53(+308%)	0.59(-29%)
Qwen$_s$→Med42	288	2040	0.12	0.83	222(-23%)	162(-92%)	0.58(+383%)	0.62(-25%)
Qwen→Qwen$_s$	277	1241	0.18	0.76	216(-22%)	159(-87%)	0.58(+222%)	0.58(-24%)
Qwen→Med42	297	2060	0.13	0.86	229(-23%)	182(-91%)	0.56(+331%)	**0.64**(-26%)
Qwen$_s$→Llama	**309**	1970	0.14	0.84	**244**(-21%)	170(-91%)	0.59(+321%)	**0.64**(-24%)

Table 2: Results from best configurations between wo/ judge (left) and w/ judge (right) configuration, and w/o rewriter (top) and w/ rewriter (top) configuration. We report the number of true positives (TP), false positives (FP), precision (P), and recall (R). Green cells indicate improvement with respect to the w/o judge configuration. Best results in **bold**.

achieve the highest performance with the Qwen model, reaching a macro-F1 of 0.5046. Particularly, general-domain LLMs consistently outperform the domain-specific model (MedLlama-Med42). Moreover, the setting without a rewriter (only extractor and judge) obtains the second-best results in terms of macro-F1 with a score of 0.6293, highlighting the impact of the *LLM as a judge* in this task.

Among the settings that include a rewriter but no judge, we observe a general trend of lower performance compared to the baselines, with a best macro-F1 of 0.4496 achieved by Qwen→Qwen$_s$ (where Qwen is the rewriter and Qwen$_s$ the extractor). When the full pipeline is used, our proposed system outperforms previous approaches, achieving a best macro-F1 of 0.6445 with Qwen$_s$→Llama using *Qwen as a judge*. This substantial improvement suggests that jointly, the judge and rewriter components play a critical role.

The submitted system was based on a 3-shot setup and exhibited a trend consistent with development set findings, removing the *LLM as a judge* led to a drop in macro-F1 from 0.48 to 0.45.

5.1. Ablation Study

We further analyze the impact of the *LLM as a judge* and *LLM as rewriter* components on the development set performance. We rely on additional metrics for further examination of our approach. Table 2 compares the configurations with and without these LLM components across all models.

First, studying the effect of the rewriter, models with an *LLM as rewriter* achieve, on average, higher true positives (272.5 vs 289 w/o judge and 214.3 vs 222.8 w/ judge) and higher recall (0.79 vs 0.82, and 0.58 vs 0.61 for configuration w/o and w/ judge, respectively). These results suggest that rewriting effectively leverages LLM parametric knowledge to extract correct factual information. However, the expansion of the input context also leads to a substantial increase in false positives, i.e. cases where the correct value is *unknown* but the *LLM as an extractor* incorrectly assigns a value. This increase in false positives penalizes the macro-F1 score.

Second, analyzing the effect of *LLM as a judge*, our results show that applying this component consistently reduces true positives by at most 30%, yet achieves a reduction in false positives up to 92%, resulting in higher precision across all configurations. This suggests that *LLM as a judge* promotes a more conservative prediction strategy, improving the faithfulness of the extracted answers to the source text, which is a desirable behavior in sensitive domains such as healthcare.

6. Conclusion

In this paper, we described our system approach for the automated CRF filling from clinical notes, in the context of the CRF Filling 2026 shared task. Our framework combines three open-weight LLMs containing: an *LLM as rewriter*, an *LLM as an extractor*, and an *LLM as a judge*, the latter responsible for filtering out predictions unsupported by the clinical note. Our analysis showed that rewriting clinical notes prior to extraction effectively leverages the parametric knowledge of LLMs, recovering a higher number of correct factual values. However, this comes at the cost of increased false positives, as the expanded context encourages the model to assign values even when the answer should be

unknown. The *LLM as a judge* addresses this limitation by adopting a conservative prediction strategy, improving precision while reducing the true positives. Our complete framework achieves a score of 0.64 in terms of macro-F1 on the development set, confirming the complementary role of each component.

7. Acknowledgements

This work was supported by the PARTAGES project, winner of the Bpifrance France 2030 call for proposals “Digital Commons for Generative Artificial Intelligence". The present work was performed using computing resources of CRIANN (Normandy, France).

8. Bibliographical References

Shantala Bellary, Binny Krishnankutty, and MS Latha. 2014. Basics of case report form designing in clinical research. *Perspectives in clinical research*, 5(4):159–166.

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

Pietro Ferrazzi, Soumitra Ghosh, Alberto Lavelli, and Bernardo Magnini. 2026. Overview of the CRF 2026 shared task on clinical case report forms filling. In *Proceedings of the Third Workshop on Patient-Oriented Language Processing (CL4Health)*, Palma, Mallorca (Spain). ELRA.

Pietro Ferrazzi, Alberto Lavelli, and Bernardo Magnini. 2025. Converting annotated clinical cases into structured case report forms. In *Proceedings of the 24th Workshop on Biomedical Language Processing*, pages 307–318, Vienna, Austria. Association for Computational Linguistics.

Alba Gutiérrez-Sacristán, Simran Makwana, Audrey Dionne, Simran Mahanta, Karla J Dyer, Faridis Serrano, Carmen Watrin, Pierre Pages, Sajad Mousavi, Anil Degala, et al. 2024. Development and validation of an open-source pipeline for automatic population of case report forms from electronic health records: a pediatric multicenter prospective study. *EBioMedicine*, 108.

Gabriela Anna Kaczmarek, Pietro Ferrazzi, Lorenzo Porta, Vicky Rubini, and Bernardo Magnini. 2026. Toward automatic filling of case report forms: A case study on data from an italian emergency department.

Jesus Lovon-Melgarejo, Jose G. Moreno, Christine Damase-Michel, and Lynda Tamine. 2026. Retop: Learning to rewrite electronic health records for clinical prediction. WSDM ’26, page 458–468, New York, NY, USA. Association for Computing Machinery.

William R Mac Kenzie, Arthur J Davidson, Andrew Wiesenthal, Jeffrey P Engel, Kathryn Turner, Laura Conn, Scott J Becker, Sharon Moffatt, Samuel L Groseclose, Jim Jellison, et al. 2016. The promise of electronic case reporting. *Public Health Reports*, 131(6):742–746.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2463–2473.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 744–756, Dublin, Ireland. Association for Computational Linguistics.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets LLM: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Zhenbang Wu, Anant Dadu, Mike Nalls, Faraz Faghri, and Jimeng Sun. 2024. Instruction tuning large language models to understand electronic health records. *Advances in Neural Information Processing Systems*, 37:54772–54786.

Ran Xu, Wenqi Shi, Yue Yu, Yuchen Zhuang, Bowen Jin, May Dongmei Wang, Joyce Ho, and Carl Yang. 2024. RAM-EHR: Retrieval augmentation meets clinical predictions on electronic health records. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 754–765, Bangkok, Thailand. Association for Computational Linguistics.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng

Ji, and Tong Zhang. 2024. R-tuning: Instructing large language models to say 'I don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7113–7139, Mexico City, Mexico. Association for Computational Linguistics.