Innov8rs at CRF Filling 2026: An Iterative Multi-LLM Ensemble Pipeline with Dynamic Few-Shot Retrieval and Data-Driven Precision Filtering

Samminga Sainath Rao[1], **Sumit Mishra**[2], **Chanchal Suman**[3]
[1]Dept. of Computer Science and Engineering, RGIPT, Amethi, India
[2]Dept. of Computer Science and Engineering, NIT Warangal, Warangal, India
21cs2018@rgipt.ac.in, sumit@nitw.ac.in, chanchal@nitw.ac.in

Abstract

In this paper, we present the technical report on the CL4Health 2026 Shared Task on Case Report Form (CRF) filling for our team Innov8rs. The paper explains the complete development of our system for the CL4Health 2026 Shared Task. We describe every phase of our system – from initial catastrophic failures with small models producing over $4,800$ false positives, through prompt engineering breakthroughs, to our final multi-LLM ensemble combining Gemini 2.5 Flash and Llama 3.3 70B with dynamic TF-IDF-based few-shot retrieval. The main contribution of this work is a data-driven precision filter that suppresses predictions for CRF items with historically high false-positive rates. This single intervention reduced false positives from 816 to 171 on the English development set, boosting macro-F1 from 0.541 to 0.703. We document the engineering challenges of multi-API-key rotation across 11 Google API keys and 2 Groq keys, the design of four distinct ensemble strategies, and the critical analysis of why development-calibrated filters suffered from distribution shift on test data (final test F1: 0.47).

Keywords: Large Language Model, Few-Shot Prompting, Clinical NLP, Information Extraction, Hallucination Suppression, Model Ensembling, Precision Filtering

1. Introduction

The automated extraction of structured clinical variables from unstructured patient notes is a cornerstone of modern medical informatics, enabling secondary use of health data for epidemiological research, quality improvement, and clinical trial screening. The CL4Health 2026 Shared Task on Case Report Form (CRF) filling presents a particularly challenging scenario: participants must predict the value of 134 discrete medical items – spanning binary conditions, categorical states, vital sign interpretations, and free-text laboratory measurements – from Italian and English emergency department clinical notes documenting patients presenting with dyspnea (Ferrazzi et al., 2026b; Kaczmarek et al., 2026; Ferrazzi et al., 2025, 2026a).

The main challenge of this task is information *absence*. In any individual clinical note, the vast majority of the 134 CRF items are never mentioned. The correct prediction for an unmentioned item is "unknown". This creates an extreme class imbalance problem: the ground truth is dominated by "unknown" values, and the scoring metric (macro-averaged F1 across exact string matches) *heavily penalizes* false positives – predictions of a definite value when the ground truth is "unknown".

Large Language Models (LLMs) (Lee et al., 2020), while remarkably capable at extracting explicitly stated facts, exhibit a persistent and dangerous bias: they *hallucinate affirmative answers* (Xu et al., 2024; Maynez et al., 2020). When prompted to determine whether a patient has epilepsy from a note that never mentions epilepsy, LLMs consistently predict "n" (No) rather than "unknown". This behavior, which we term the **completion bias**, is rooted in the instruction-tuning objective that rewards helpful, complete responses.

For example, in case of clinical information extraction, this bias is catastrophic. For the item *pregnancy*, the correct answer is "unknown" (the note says nothing about pregnancy). However, every small model we tested predicted "n" (No), inferring a negative from absence. For *SpO2*, the correct answer is "95%" (the actual value), but weaker models predicted "measured" (a valid but less informative option). For *heart rate*, the model must interpret the raw number 110 against clinical thresholds and predict "tachycardic" (HR > 100).

Our contributions include: (1) a multi-provider extraction framework supporting OpenAI, Google, Anthropic, and Ollama; (2) a Schema Registry implementing strict constrained decoding against 134 valid option sets; (3) dynamic few-shot example retrieval using TF-IDF cosine similarity over a $2,667$-note silver corpus; (4) a multi-API-key rotation framework managing 11 keys for throughput maximization; (5) four distinct ensemble strategies across model outputs; and (6) a data-driven precision filter achieving our breakthrough result.

Rest of the paper describes task description and data normalization in section 2. In section 3, we describe system architecture. The precision filter is discussed in section 4, and ensemble strategies is discussed in section 5. In section 6, the analysis of results is discussed, and finally in section 7, we

Proceedings of the Third Workshop on Patient-Oriented Language Processing (CL4Health), pages 390–394
12 May 2026.

conclude the paper.

2. Task Description and Data Normalization

Given a clinical note d from an emergency department visit, the system must output predictions for all $N = 134$ predefined CRF items. Each item i has a fixed set of valid values V_i and a special value "unknown".

2.1. Schema Analysis and Constrained Normalization

We built a **Schema Registry** by parsing the provided `valid_options_train.json`. The module classified items into five types based on their valid option sets:

- **Binary** (e.g., *epilepsy*, *pregnancy*): {"y", "n"}. Highest false-positive risk because LLMs default to "n" for unmentioned items.
- **Categorical** (e.g., *heart rate*): {"tachycardic", "bradycardic", "normocardic"}. Requires medical reasoning from raw numbers.
- **Chronic conditions** (e.g., *chronic pulmonary disease*): {"certainly active", "possibly active", "resolved"}.
- **Free-text/numeric** (e.g., *SpO2*, *creatinine*): Any numeric string ("95%", "1.2"). Must extract actual values, not placeholders.
- **Lab status**: Both "measured" and actual values valid.

The normalization layer performed case-insensitive matching against valid options. Predictions not matching any valid option were coerced to "unknown", preventing format-based scoring penalties.

Algorithm 1 Dynamic Few-Shot Example Selection

Offline: Index silver standard notes $S = \{s_1, \ldots, s_M\}$ with TF-IDF (max $10,000$ features, English stop words removed)
Compute TF-IDF matrix $\boldsymbol{T} \in R^{M \times F}$
for each test note d **do**
  $\boldsymbol{q} \leftarrow$ TF-IDF(d) ▷ Query vector
  sim$_j \leftarrow \cos(\boldsymbol{q}, \boldsymbol{T}_j) \quad \forall j \in [1, M]$
  top3 $\leftarrow$ argsort(sim)$[-3:]$
  Inject $\{s_j : j \in$ top3$\}$ with their silver annotations as few-shot examples into prompt

3. System Architecture

Our final system consisted of four sequential stages each developed iteratively through failure analysis. At first, TF-IDF retrieval is done, where top-3 silver notes are used. After the retrieval, V3 system prompt and dynamic few shot example is done. Output is extracted through LLMs, schema normalization is done. Using the conservative strategy, the ensemble merging is done. Finally the precision filter is done.

3.1. Prompt Engineering: Three Iterations

Our prompts evolved through three major versions, each addressing specific failure modes.

V1 (Baseline): A minimal system prompt instructing the LLM to extract CRF values. No guidance on handling missing information. Result: massive hallucination ($> 4,800$ FP on 80 patients).

V2 (Open World Assumption): Added explicit instructions: *"'unknown' is the default value. Use when the text doest not mention the item at all. Do not assume 'n' just because an item is missing."* Result: FP dropped 80% (to 916) but remained too high.

V3 (Targeted FP Suppression + Clinical Rules): Our best prompt (69 lines) incorporated three innovations: (a) *Explicit error examples* for the top 9 hallucinated binary items (pregnancy, trauma, thoracentesis, homelessness, pacemaker, CPR, chest pain, pneumothorax), each with the instruction: *"If not mentioned, output 'unknown', NOT 'n'."* (b) *Lab value extraction rules* forbidding the generic "measured" placeholder: *"For SpO2: Extract the actual percentage value (e.g., '95%'). Do not output 'measured'."* (c) *Vital sign interpretation thresholds*: BP $\geq 140/90 \rightarrow$ "hypertensive", HR $> 100 \rightarrow$ "tachycardic", Temp $> 38°$C $\rightarrow$ "hyperthermic", RR $> 20 \rightarrow$ "tachypneic".

3.2. Dynamic Few-Shot Retrieval

Static few-shot examples cannot represent the diversity of $2,667+$ clinical presentations (Brown et al., 2020; Rubin et al., 2022). We implemented retrieval-augmented in-context learning for English: for **Italian**, the silver standard IDs did not align with the clinical notes corpus. We selected three static examples from the gold and synthetic training sets, prioritizing notes with the most diverse positive predictions.

3.3. Multi-API-Key Rotation

Processing 200 test notes with prompts exceeding $6,000$ tokens triggered severe rate limiting. We engineered a `KeyRotator` class managing 11 Gemini (Team et al., 2023) and 2 Groq keys:

In practice, all 11 Gemini keys exhausted within $15 - 20$ minutes, yielding $\sim$50 test notes per cycle. When Groq's `llama-3.3-70b-versatile`

Algorithm 2 Multi-Key Rotation Strategy

Initialize clients $C = \{c_1, \ldots, c_{11}\}$, exhausted set $E = \emptyset$
for each test note d **do**
  while extraction not successful **do**
    if $|E| = |C|$ **then**
      Wait 60s, reset $E \leftarrow \emptyset$
    $k \leftarrow$ next key $\notin E$
    result $\leftarrow$ EXTRACT(c_k, d, prompt)
    if HTTP 429 or quota error **then**
      $E \leftarrow E \cup \{k\}$, **CONTINUE**
    else
      Save result, **BREAK**

hit its 100K TPD limit, the script parsed the suggested retry time from the error message (e.g., *"try again in 38m15s"*) and slept accordingly.

4. The Precision Filter: Our Key Contribution

The 816 FP from Gemini 2.5 Flash were not randomly distributed. Computing per-item precision:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \tag{1}$$

revealed a stark bimodal distribution: **34 items** had precision ≥ 0.33 (reliably extracted), while **100 items** had precision < 0.33 (dominated by false positives). Many items had precision of exactly 0.0 – *every single prediction was wrong*. For example:

- *Abdominal pain*: TP $= 0$, FP $= 23$, Precision $= 0.0$
- *Phlebotomy*: TP $= 0$, FP $= 18$, Precision $= 0.0$
- *Respiratory failure*: TP $= 0$, FP $= 29$, Precision $= 0.0$

These items represented conditions that the LLM *always* inferred from indirect contextual clues (e.g., predicting "respiratory failure" whenever a patient presented with dyspnea) but were almost never explicitly confirmed in the ground truth.

Threshold τ	Items Kept	TP	FP	F1
0.0 (no filter)	134	352	816	0.541
0.1	62	332	298	0.654
0.2	49	321	226	0.672
$\mathbf{0.33}$	$\mathbf{34}$	$\mathbf{305}$	$\mathbf{171}$	$\mathbf{0.703}$
0.4	29	286	129	0.689
0.5	24	259	96	0.668

Table 1: English Dev F1 Across Precision Filter Thresholds.

The best threshold was found to be $\tau = 0.33$ (Table 1). Using this threshold, the filter removed 645 false positives (reducing them from 816 to 171) while losing only 47 true positives (from 352 to 305). As a result, the macro-F1 score increased from 0.541 to 0.703, which corresponds to a 30% relative improvement achieved through a single post-processing step.

A similar trend was observed on the Italian development set. Applying the same method improved the F1 score from 0.500 to 0.683, reducing 640 false positives while losing only 39 true positives.

Table 2 isolates the contribution of each component.

Config.	TP	FP	FN	F1
English Dev				
V1 + Flash Lite	317	4814	22	0.355
+ V2 Prompt	280	916	55	0.465
+ V3 Prompt	306	616	52	0.496
+ 2.5 Flash	352	816	17	0.541
+ Prec. Filter	305	171	83	$\mathbf{0.703}$
Italian Dev				
V1 + Flash Lite	315	6006	13	0.302
+ V2 Prompt	261	1584	77	0.434
+ 2.5 Flash	318	802	14	0.500
+ Prec. Filter	279	162	108	$\mathbf{0.683}$

Table 2: Cumulative Contribution of Each Component

The precision filter alone accounts for more F1 improvement $(+0.162)$ than model upgrade $(+0.045)$, prompt V3 $(+0.031)$, and prompt V2 $(+0.110)$ combined.

5. Ensemble Strategies for Test Submission

For the test set, we had Gemini 2.5 Flash predictions for $49/200$ notes and Llama 3.3 70B for $200/200$. We designed four ensemble strategies:

S1 – Consensus: Keep a prediction only when both models give the same non-"unknown" answer. If they disagree, the output is set to "unknown". This approach focuses on improving precision by requiring agreement between the two models.
S2 – Groq-Only: Use predictions only from Llama 3.3 70B. This acts as a baseline with full coverage.
S3 – Gemini-First: Use Gemini predictions when they are available (for 49 notes), and use Groq predictions for the remaining 151 notes. This strategy gives priority to the stronger model when possible.
S4 – Conservative (Best Performer): For notes where both models provide predictions, if Gemini outputs "unknown", the final result is also set to "unknown", even if Groq predicts something else. If both models give non-"unknown" predictions, Gemini's prediction is used. For notes where Gemini

predictions are not available, Groq's predictions are used as they are.

Test Submission	Macro-F1
S2: Groq-Only	0.46
S3: Gemini-First Merge	0.45
S4: Conservative Ensemble	$\mathbf{0.47}$
Pure Gemini ($49/200$ coverage)	0.47
Conservative + Precision Filter	0.46

Table 3: Test set scores across submission strategies.

The Gemini-First merge (S3) scored *lower* than Groq-Only (S2), suggesting that Gemini introduced false positives on the notes it covered. The Conservative strategy (S4) mitigated this by using Gemini primarily as a filter rather than a source.

6. Analysis and Discussion

The most significant finding is the gap between development (0.703) and test (0.47) performance. More counterintuitively, applying the dev-calibrated precision filter to test predictions *decreased* the score from 0.47 to 0.46.

This exposes a fundamental limitation of post-hoc data-driven filtering: the 100 items blacklisted (based on dev precision) may represent genuine clinical findings present in Test patients but absent from the 80 dev patients. With only 80 calibration samples, per-item precision estimates are inherently unstable. Items with 0 TP and 5 FP on the Dev set might have 3 TP and 2 FP on the Test set – but our filter would suppress all 3 true positives.

6.1. Model Scaling Effects

A notable observation: the worst models produced the *most* non-unknown predictions ($\sim$60/note for Flash Lite) while the best models produced the fewest (6–10/note for Gemini 2.5 Flash). This inverse relationship between extraction density and quality confirms that **conservatism is the dominant strategy** in sparse extraction tasks under macro-F1 evaluation.

6.2. Key Takeaways

From the results, it is noticed that scaling model in terms of parameters, but is not always sufficient. It is also seen that, the precision filter ($+0.162$ F1) contributed more than all prompt iterations combined ($+0.141$ F1). However, it requires representative calibration data. With < 100 calibration patients, item-level statistics do not generalize robustly.

7. Conclusion

The CRF-filling experiment demonstrated that modern LLMs exhibit a strong completion bias when extracting sparse clinical information, requiring significant engineering efforts to mitigate it. Starting from an initial baseline that produced 4,814 false positives, our iterative pipeline—incorporating prompt engineering, larger model variants, dynamic retrieval mechanisms, and statistical filtering—reduced development false positives by 96% and improved the English development macro F1 score from 0.355 to 0.703. However, the discrepancy between development performance (0.703) and test performance (0.47) indicates the importance of more distribution-robust post-processing strategies. Future research should investigate: (1) instruction fine-tuning approaches, such as DPO or RLHF, that explicitly encourage an "unknown" response preference to address completion bias at the model level; (2) item-level confidence calibration based on generation probabilities; and (3) cross-validated precision estimation using bootstrap sampling to derive more reliable and generalizable item-level filtering methods.

8. Bibliographical References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Pietro Ferrazzi, Mattia Franzin, Alberto Lavelli, and Bernardo Magnini. 2026a. Small LLMs for Medical NLP: A Systematic Analysis of Few-Shot, Constraint Decoding, Fine-Tuning and Continual Pre-Training in Italian.

Pietro Ferrazzi, Soumitra Ghosh, Alberto Lavelli, and Bernardo Magnini. 2026b. 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, Viena, Austria. Association for Computational Linguistics.

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.

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

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1906–1919.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2655–2671.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.