Actes de CORIA-TALN-RJCRI-RECITAL 2025
Marseille, France, 30 juin au 4 juillet 2025

Structuration Automatique de la Posologie en Français : Quel rôle pour les LLMs ?

Natalia Bobkova[1] Laura Zanella-Calzada[1] Anyes Tafoughalt[1,2]
Raphaël Teboul[1] François Plesse[1] * Félix Gaschi[1]
(1) SAS Posos, (2) Sorbonne Université, Paris, France
felix@posos.co

RÉSUMÉ

La structuration automatique de posologie est essentielle pour fiabiliser la médication et permettre une assistance à la prescription médicale. Les textes de prescriptions en français présentent très souvent des ambiguïtés, des variabilités syntaxiques, et des expressions colloquiales, ce qui limite l'efficacité des approches classiques de machine learning. Nous étudions ici l'emploi de Grands Modèles de Langages (*LLM*) pour structurer les textes de posologie en comparant des méthodes fondées sur le *prompt-engineering* et le *fine-tuning* de LLM avec un système "pré-LLM" fondé sur un algorithme de reconnaissance et liaison d'entités nommées (NERL). Nos résultats montrent que seuls les LLM fine-tunés atteignent la précision du modèle de référence. L'analyse des erreurs révèle une complémentarité des deux approches : notre NERL permet une structuration plus précise, mais les LLMs captent plus efficacement les nuances sémantiques. Ainsi, nous proposons le modèle hybride suivant : faire appel à un LLM en cas de faible confiance en la sortie du NERL (<0.8) selon notre propre score de confiance. Cette stratégie nous permet d'atteindre une précision de 91% tout en minimisant le temps de latence. Nos résultats suggèrent que cette approche hybride améliore la précision de la structuration de posologie tout en limitant le coût computationnel, ce qui en fait une solution scalable pour une application clinique en conditions réelles.

ABSTRACT

Automatic Posology Structuration : What role for LLMs ?

Automatically structuring posology instructions is essential for improving medication safety and enabling clinical decision support. In French prescriptions, these instructions are often ambiguous, irregular, or colloquial, limiting the effectiveness of classic ML pipelines. We explore the use of Large Language Models (LLMs) to convert free-text posologies into structured formats, comparing prompt-based methods and fine-tuning against a "pre-LLM" system based on Named Entity Recognition and Linking (NERL). Our results show that while prompting improves performance, only fine-tuned LLMs match the accuracy of the baseline. Through error analysis, we observe complementary strengths : NERL offers structural precision, while LLMs better handle semantic nuances. Based on this, we propose a hybrid pipeline that routes low-confidence cases from NERL (<0.8) to the LLM, selecting outputs based on confidence scores. This strategy achieves 91% structuration accuracy while minimizing latency and compute. Our results show that this hybrid approach improves structuration accuracy while limiting computational cost, offering a scalable solution for real-world clinical use.

MOTS-CLÉS : TAL, Larges Modèles de Langage, TAL clinique, TAL médical.

KEYWORDS: NLP, Large Language Models, clinical NLP, medical NLP.

∗. Was at Posos at the time of his contribution.

Actes de l'atelier Traitement du langage médical à l'époque des LLMs 2025 (MLP-LLM), pages 49–67.

1 Introduction

Structuring posology instructions is essential for enhancing medication safety and enabling clinical applications, including decision support systems (Elhaddad & Hamam, 2024). In French prescriptions, posologies are frequently expressed in diverse, ambiguous, and sometimes colloquial language, which complicates their automatic structuration.

Traditional information extraction pipelines, typically based on Named Entity Recognition (NER) and rule-based post-processing, offer reliable performance but struggle with linguistic variability and unexpected formulations (Liang *et al.*, 2019; Gaschi *et al.*, 2023). In contrast, Large Language Models (LLMs) have recently shown strong potential for medical text understanding (Hu *et al.*, 2024; Biana *et al.*, 2024), though their application to specialized tasks like posology structuration remains largely unexplored.

In this work, we investigate the use of LLMs to structure French posology instructions into standardized formats. We benchmark several open-source and proprietary LLMs using prompt engineering techniques such as chain-of-thought, few-shot, and contrastive prompting. To further improve performance, we apply lightweight fine-tuning on synthetic posology data.

Our results show that while prompt engineering narrows the gap, fine-tuning is necessary for LLMs to match the performance of our robust internal baseline (based on Named Entity Recognition and Linking, or NERL). Through detailed error analysis, we highlight complementary strengths between rule-based systems and LLMs. Finally, we propose a hybrid strategy that leverages model confidence scores to combine their outputs, offering a practical and efficient solution for real-world deployment.

We summarize our contributions as follows :

1. We present a novel dataset, MEDPOSOSF , for evaluating LLM-based structuration of French posology instructions, extending beyond standard NER by requiring full structured outputs.
2. We benchmark a range of LLMs, open and proprietary, using prompt engineering and fine-tuning, and analyze their ability to handle the linguistic variability of real-world prescriptions.
3. We propose a confidence-based hybrid pipeline that combines a rule-based NERL system with LLMs, achieving improved accuracy (91%) while controlling latency and resource usage.

2 Related Work

Research on structuring clinical texts often begins with simpler tasks such as NER, which serves as a foundational step for more complex transformations, including the structuring of dosage instructions.

Named Entity Recognition in the Medical Domain NER is a core task in biomedical Natural Language Processing (NLP), aiming to identify entities such as medications, dosages, symptoms, or diseases within clinical texts. Over time, several approaches have been developed, each attempting to overcome the limitations of the previous ones.

Early methods were mainly based on handcrafted rules and domain-specific dictionaries. These systems typically relied on keyword selection and pattern-matching techniques to extract relevant terms from text (Krstev *et al.*, 2014). While they offered high precision in specific contexts, their

recall was often limited, and they lacked adaptability. Moreover, maintaining and updating these resources required significant effort and deep domain expertise (Chen *et al.*, 2020), making them difficult to scale across different datasets or languages.

To address these issues, statistical machine learning models like Hidden Markov Models (HMMs) (Shen *et al.*, 2003) and Conditional Random Fields (CRFs) emerged (He & Wang, 2008; Settles, 2004; Jiang *et al.*, 2011). CRFs, in particular, became widely used for sequence labeling tasks, as they modeled label dependencies and improved prediction consistency. In the French context, Bigeard *et al.* (2015) proposed a hybrid system combining CRFs and handcrafted rules to extract numerical values from unstructured EHRs. Their work highlighted challenges specific to the French language, such as ambiguous units and linguistic variability, emphasizing the need for language-specific approaches in biomedical NLP.

However, they still depended heavily on feature engineering and struggled to capture long-distance relationships between tokens. A major shift came with the introduction of deep learning. Long Short-Term Memory (LSTM) networks allowed models to retain information across longer sequences (Peng & Dredze, 2017; Li *et al.*, 2016). Their bidirectional versions, BiLSTMs, further improved context modeling by processing text in both forward and backward directions (Wang *et al.*, 2015). The combination of BiLSTM with CRFs led to the well-known BiLSTM-CRF architecture, which merged contextual encoding with structured prediction (Huang *et al.*, 2015; Lample *et al.*, 2016). This hybrid approach became a reference in the field, especially when combined with domain-adapted embeddings such as those trained on PubMed (e.g., Word2Vec or GloVe). However, the use of static embeddings still limited the model's ability to adapt to varying contexts.

More recently, the use of self-attention mechanisms (Vaswani *et al.*, 2023) and transformer-based architectures has significantly improved performance in NER. Unlike RNNs, self-attention can model global dependencies in a single pass, making it especially well-suited for capturing the structure of clinical narratives.

In this context, transformer-based models like BERT (Devlin *et al.*, 2019) and its biomedical variants have set new standards for medical NER. BioBERT (Lee *et al.*, 2019), pretrained on large biomedical corpora, ClinicalBERT (Alsentzer *et al.*, 2019), fine-tuned on clinical notes (MIMIC-III), and SciSpacy (Neumann *et al.*, 2019), built on PubMed articles, have all achieved state-of-the-art results. These models benefit from contextual embeddings tailored to biomedical language, drastically reducing the need for manual feature design and improving robustness in handling ambiguous or complex terms.

Using Large Language Models (LLMs) for Medical NER With the rise of general-purpose Large Language Models (LLMs) like GPT-3.5 and GPT-4, researchers have started exploring their use for biomedical NER. Hu *et al.* (2024) evaluated GPT-3.5 and GPT-4 on two clinical NER tasks : identifying medical problems, tests, and treatments from clinical notes, and extracting adverse events from safety reports. Under this setup, GPT-4 performance is close to those of domain-specific models like BioClinicalBERT.

To improve performance, some researchers have proposed adapting LLMs more explicitly to NER. For instance, Wang *et al.* (2023) introduced GPT-NER, a framework that reframes NER as a text generation task. This allows the model to directly output labeled sequences, rather than relying on token classification. Their results suggest that this generation-based formulation can be especially effective in low-resource scenarios, where annotated data is scarce.

More recently, Biana *et al.* (2024) presented VANER, a system built on top of LLaMA2. It combines domain-specific instructions and external biomedical knowledge to improve entity recognition. VANER outperformed previous LLM-based methods and even surpassed some traditional BioNER baselines on multiple datasets.

In parallel, Naguib *et al.* (2024) conducted a large-scale evaluation of generative and masked LLMs for few-shot clinical NER across English, French, and Spanish. Their results show that while prompt-based models perform well on general NER, they are outperformed in the clinical domain by lighter, fine-tuned models like BiLSTM-CRF.

Overall, while general-purpose LLMs may not yet consistently outperform specialized models like BioClinicalBERT, studies show that prompt engineering and hybrid strategies (e.g., adding domain knowledge) can significantly boost their effectiveness for clinical NER.

Structuring Posology Instructions : Beyond NER While NER focuses on identifying relevant entities in text, dosage structuring advances further by transforming complex dosage instructions into structured formats (e.g., JSON), facilitating integration into clinical information systems.

A key milestone in the structuration of medication-related data was the i2b2 Medication Challenge (Uzuner *et al.*, 2010), which defined subtasks such as drug name, dosage, frequency, and route extraction. It remains a standard benchmark for evaluating medication information extraction systems, particularly in English clinical texts.

So far, only a few studies have looked into applying LLMs to tackle related tasks. For instance, Van *et al.* (2024) introduced Rx Strategist, a multi-agent LLM pipeline augmented with knowledge graphs and search strategies to verify prescriptions by analyzing indications, dosages, and potential drug interactions. This approach achieved performance comparable to that of experienced pharmacists. Similarly, Isaradech *et al.* (2024) investigated zero and few-shot NER and text expansion in medication prescriptions using ChatGPT 3.5 and highlighting the potential of LLMs in handling varied and ambiguous language in prescriptions.

Silva & Gomes (2025) developed an adaptive LLM-based intelligent medication assistant aimed at supporting decision making in antidepressant prescriptions, demonstrating the utility of fine-tuned language models in clinical decision support workflows and emphasizing the importance of structured dosage information for clinical decision making. Haaker *et al.* (2025) conducted a comparative evaluation of five approaches for extracting daily dosages, including parsing techniques, LLMs (GPT-4o), and the rule-based system RxSig, reporting slightly better performance with GPT-4o than with RxSig, along with notable differences in sensitivity and computational demands.

These studies highlight the growing role of LLMs in transforming how posology instructions are interpreted and structured, offering richer and more flexible representations than traditional NER approaches.

Our work extends this research approach to French prescriptions, marking, to our knowledge, one of the first efforts to apply LLMs to the structured extraction of posology instructions.

3 Methodology

3.1 The posology structuration task

We release **Med**ical **Poso**logy **S**tructuration in **F**rench (MEDPOSOSF), a posology structuration test dataset for converting free-text posology instructions into structured JSONs, through which we normalize elements of the posology instruction like the frequency, the intake dose, or the duration of the treatment.

We collected sentences containing dosage instructions from a private set of French scanned typewritten prescriptions. Following Gaschi *et al.* (2023), those prescriptions were passed through an Optical Character Recognition (OCR) system [1]. Only the sentences containing posology instructions and no other patient information were kept, ensuring anonymization of the data [2].

Those sentences were then manually annotated by medical experts. Contrary to simpler structuration tasks like NER, the task is not a classification, but is more akin to a text-to-JSON conversion. Thus, instead of tagging spans of text with relevant labels, annotators were asked to fill a form for each relevant spans of text (e.g. "1 ampoule tous les 2 mois pendant 4 mois"), leading to a JSON with the following important fields (The complete schema is shown in Appendix B) [3] :

- `quantity_and_rate` : dict, contains the amount of intake units to take at once
  - `value` : float, the amount of unit to take at once (1.0 in our example)
  - `unit` : str (optional), the type of unit ("ampoule(s)")
  - `code` : str (optional), the SNOMED code for the unit ("413516001")
- `timing` : dict, contains the timing with which each intake must take place
  - `frequency` : int, an integer for the number of intake for the given period (1)
  - `period` : int, the number of period_unit to observe before repeating the intake (2)
  - `period_unit` : str, the time unit used for measuring the period between intakes ("month")

Our evaluation metric is a query-wise exact accuracy, which measures the proportion of queries for which each field is correctly predicted [4]. We publicly release our manually annotated dataset and code for evaluating the predictions of any model [5].

3.2 Baseline : an internal pre-LLM system

We design our baseline as a hybrid system that combines transformer-based and rule-based NER with embedding-based Named Entity Linking (NEL). The NER module consists of a fine-tuned `microsoft/mdeberta-v3-base` (He *et al.*, 2023) model. It is trained on an internal manually

1. https://cloud.google.com/vision
2. To ensure anonymization, we manually checked each sentence in the final dataset and removed any that contained personal info (name, social security number, doctor ID)
3. Annotations were performed by two annotators using an internally-developped tool that converts the output of an user-friendly form into JSON.
4. SNOMED codes were not considered in the evaluation, although some LLMs are sometimes able to find the right code, we leave this named entity linking step for future work.
5. https://github.com/posos-tech/posology-structuration-task

annotated dataset [6], which we unfortunately cannot release due to issues of confidentiality. This dataset contains 1,699 sentences annotated with 5,845 entities (Full list of entities in Appendix D).

Some entity types are rare, for example, there are only 19 entities of type `TIME_OF_DAY` (like "à 08h") in the entire training set. This requires some form of data augmentation. We thus define a set of rules to generate synthetic training examples that include those rarer entities.

For some of the entities detected with the NER, we need to link them to relevant concepts using NEL. We link recognized dose units to normalized concepts from the SNOMED CT terminology. Our linking pipeline embeds candidate concepts using `FastText` (Joulin *et al.*, 2016) word vectors aggregated into sentence-level representations. We retrieve the closest concepts based on cosine similarity.

However, extracting and linking entities is not sufficient to produce an arborescent JSON describing the posology. An internal rule-based post-processing system allows to combine the detected entities into a structured JSON. We choose not to disclose our entire set of rules, as they were developped internally and iteratively across several years. A typical example of such rules is that, in "3 cp le matin" two entities are initially detected : a dose (3 cp) and an entity of type `WHEN` (le matin). The rule-based system will then combine them into a single JSON object with the `quantity_and_rate` filled with the information from the dose and the `frequency` (1), the `period` (1) and the `period_unit` (day) are inferred from the `WHEN` entity.

3.3 Confidence score

Following the method proposed by Lahoti *et al.* (2021), we crafted a confidence score aimed to assess risk, defined by three kinds of uncertainties : model, aleatoric and epistemic. Considering the structuration model, i.e. the NERL baseline, as a black-box classification model, we trained an ensemble of Gradient Boosted classifiers as a meta-learner to estimate these uncertainties. Each model is a regression tree that outputs a score for the correct or incorrect label. The final prediction is obtained by averaging the outputs of all models in the ensemble. The different uncertainties were evaluated as follow :

- **Model uncertainty** : out of NERL model information, estimated using the final model score.
- **Aleatoric uncertainty** : variability of data points, and noise caused in our case by mistakes of OCR for example, evaluated by averaging the entropy of each smaller model.
- **Epistemic uncertainty** : systematic gaps in the training samples, it mostly arises when outliers are present within the training dataset. It was deduced from the total uncertainty, estimated from the entropy of the model outputs.

We trained the confidence score on two datasets with a total of 600 sentences : each sentence is paired with the JSON output produced by our NERL baseline and manually annotated as "correct" or "incorrect" according to the ability of our model to correctly structure them. The confidence score model is then trained, using the sentence and the JSON output as features, to predict the correctness of the output.

6. For the annotation process, annotators used an internally developed tool that allows them to select relevant textual information and assign the appropriate tags.

3.4 LLM approaches

While the aforementionned pre-LLM system is effective for structuring posology instructions (see results), it is limited by its reliance on handcrafted rules, the need for extensive domain knowledge, and requires a training dataset for the NER. To address these limitations, we explore the use of Large Language Models (LLMs) to automate the structuration process, without using any handcrafted rules or training dataset other than a synthetic one.

Prompt engineering

We explore the integration of Large Language Models (LLMs) to improve the pososlogy structuration task, focusing on prompt engineering as an initial approach. Our experiments involve several open-source models, such as LLaMa 3.2 (Grattafiori *et al.*, 2024), Gemma 2 (Team *et al.*, 2024b), Mistral 7B (Albert *et al.*, 2023), and Phi 3.5 (Abdin *et al.*, 2024) ; as well as proprietary models, particularly the Gemini family (Team *et al.*, 2024a), deployed via Vertex AI.

To optimize model behavior, we employ several prompting strategies : chain-of-thought prompting (Wei *et al.*, 2022) which encourages a step-by-step reasoning ; few-shot prompting (Brown *et al.*, 2020) to embed structured examples in the prompt ; reformulation instructions (Deng *et al.*, 2023) which direct the model to clarify or rephrase the posology instruction before structuring it ; contrastive prompting (Chia *et al.*, 2023) which includes examples of common errors to help the model avoid frequent pitfalls. The prompts used are described in detail in Appendix E.

Fine-tuning with synthetic data

To further improve LLM performance, we apply lightweight fine-tuning techniques such as adapter-based methods (Houlsby *et al.*, 2019). Given the scarcity of annotated prescription data, we construct a synthetic dataset of over 1,600 posologies, structured using our NERL baseline, alleviating the need for any manual annotation [7].

From this set, approximately 1,200 are filtered out based on cofidence score (>0.7). Despite being synthetic, the dataset reflects the structural and linguistic diversity of real-world posologies and serves as an effective foundation for task-specific model tuning (Gunasekar *et al.*, 2023).

4 Experiments and Results

4.1 Models comparison

Our first objective is to select a suitable LLM for the posology structuration task. We run initial baseline tests using simple prompts across open-source (LLaMA 3.2 [8], Gemma 2 [9], Mistral 7B [10],

7. Although the baseline itself required annotated data to function
8. 3.21B parameters, Q4_K_M quatisation
9. 9.24B paramerters, Q4_0 quatisation
10. 7.25B parameters, Q4_0 quatisation

Phi 3.5 [11]) and closed-source (Gemini family) models, evaluating both their accuracy and inference latency [12] on manually curated data set (Table 1).

Model	Average Accuracy	Average Latency
Vertex AI - Gemini 1.0 Pro	38%	14 min
Vertex AI - Gemini 1.5 Pro	47%	11 min
Vertex AI - Gemini 1.5 Flash	42%	3 min
Ollama - Gemma 2	51%	20 min
Ollama - Mistral 7	28%	13 min
Ollama - Phi 3.5	22%	9 min
Ollama - LLaMA 3.2	6%	9 min

TABLE 1 – Preliminary evaluation of LLMs on the posology structuration task. The latency is measured across the whole test dataset. Due to the closed-source nature of Gemini models Ollama's and the NERL's latency is not directly comparable with Vertex AI, as it does not run on the same hardware. However, the latency still provides a useful indication of what is possible with readily available models and hardware.

From these initial results, Gemini 1.5 Flash emerges as the most promising candidate, offering a good balance between performance and latency. While Gemini 1.5 Pro and Gemma 2 are slightly outperforming in terms of accuracy, their significantly higher inference time makes them impractical for iterative experimentation.

Once the model is selected, we proceed with iterative prompt refinement. Importantly, we observe that prompt engineering is highly model-specific ; optimized prompts used for one LLM do not transfer effectively to others. Therefore, we proceed with development and evaluation of prompt strategies exclusively for Gemini 1.5 Flash [13]. This is consistent with previous literature, which shows that engineered prompts often do not transfer well across different models (Ye *et al.*, 2024).

We begin with relatively simple prompting techniques [14], including direct task instructions, chain-of-thought reasoning, and few-shot examples embedded in the prompt. While these approaches improve performance, they do not fully resolve the inherent ambiguities of medical language. To mitigate this, we introduce rephrase-and-respond instructions, guiding the model to first reformulate the input to clarify intent before attempting to structure it. Finally, we apply contrastive prompting, where examples of incorrect outputs and common mistakes were explicitly included to guide the model away from frequent failure patterns.

All prompts are executed with the temperature set to zero, ensuring deterministic and consistent output generation. We empirically observed that setting the temperature to zero led to improved accuracy, as shown in Table 2. This is consistent with previous findings that lower temperatures have a tendency to produce more repetitive and less diverse outputs, which resemble the training data more closely (Renze, 2024).

Using this fully engineered prompt set, Gemini 1.5 Flash achieves an accuracy of 73%. However, despite the gains, prompt engineering alone still falls short of our NERL baseline. The results motivate

11. 3.82B parameters, Q4_0 quatisation
12. Except for Gemini models which run on Google servers, all tests for Ollama models were performed on the same GPU
13. Specifically, we evaluated the final prompt optimized for Gemini 1.5 Flash on other models. Gemini 1.5 Pro, despite outperforming Flash on the initial prompt, achieved 8% lower accuracy than Flash when using the Flash-optimized version.
14. Further details on prompt design are provided in Appendix A

top k	top p	temperature	Average Accuracy
40	0.80	2.0	68.75
35	0.90	1.5	67.19
30	0.85	1.2	68.75
15	0.90	1.0	68.75
5	0.97	0.7	71.88
5	1.00	0.1	73.00
-	-	0.0	73.00

TABLE 2 – Evaluation of different parameter settings for the Gemini 1.5 Flash model. The parameters are : **top k** (number of top tokens to consider), **top p** (nucleus sampling threshold), and **temperature** (controls randomness in output generation).

the application of fine-tuning.

We fine-tune the model with PEFT method available throuth Vertex AI using a filtered synthetic training dataset of 1 200 high-confidence outputs generated by our NERL pipeline. The tuning is performed in two stages. First, the model is trained on plain input-output pairs, resulting in an accuracy of 79%. In the second stage, we include a partial version of the final prompt within the training inputs : general task instructions, a chain-of-thought example, and a list of mistakes to avoid. This alignment leads to a further improvement, reaching 84% accuracy and approaching our NERL baseline. Final results are shown in Table 3.

Model Configuration	Average Accuracy	Average Latency
Gemini 1.5 Flash (prompt-engineered)	73%	3 min
Gemini 1.5 Flash (fine-tuned)	79%	3 min
Gemini 1.5 Flash (prompt-engineered + fine-tuned)	84%	3 min
NERL baseline	85%	1 min

TABLE 3 – Final evaluation of LLMs on the posology structuration task. Please note that the average latency for the NERL baseline is not directly comparable to the LLMs, as it does not run on the same hardware.

In summary, our experiments demonstrate that while prompt engineering can significantly improve LLM performance on complex, ambiguous tasks like posology structuration, only model-specific fine-tuning enables LLMs to reach the performance levels of a robust baseline.

4.2 Error analysis

To compare the performance of NERL and Gemini 1.5 Flash we conduct an error analysis. Both models demonstrate high performance in structuring posologies, but diverge in their handling of linguistic complexity and ambiguity.

NERL, for instance, struggles with implicit or colloquial conditional expressions. For example, it fails to extract *si besoin* from *Lactulose 10 g : si besoin*, and similarly misses *si douleurs* as a conditional intake trigger. It also shows weakness in detecting and normalizing OCR errors, for example, *sachet* in *1 sach dose...*. Additionally, unit recognition is sometimes inconsistent, particularly when syntaxis is atypical, as seen in *Comprimé : 2 à 20 :00*, where *comprimé* was not extracted.

In contrast, Gemini tends to add unnecessary abstractions, like extracting *period : 1* and *period unit : day* from *prendre 1 comprimé 3 fois par 24 heures* or refomatting, like transforming *15/11/2022* into *2022/11/15*. It also has difficulty parsing multi-timepoint instruction : in *2 gélules au cours ou immédiatement après les repas*, it confuses overlapping timing cues (like *during the meal* vs. *after a meal*) and returns only one. Gemini also misinterpretes dose allocation across multiple intake times : *Prendre 2 gélules le matin et le soir...* was interpreted as 2 capsules once daily, instead of once in the morning and once in the evening.

In conclusion, while NERL and Gemini 1.5 Flash each show specific weaknesses, they also demonstrate complementary strengths. NERL favors precision and structural consistency, whereas Gemini excels in semantic interpretation and normalization. Used together, these models could provide a more robust and comprehensive solution for structuring posology task.

4.3 Towards an LLM-based hybrid system

Rather than replacing NERL, we propose a hybrid approach based on confidence score. The workflow we propose begins with processing the input document using NERL. If NERL's confidence score falls below a defined threshold (<0.8), the task is delegated to the LLM for supplementary analysis. The final output is selected based on the highest confidence score between the two systems.

This hybrid strategy leverages the complementary strengths of NERL and the LLM. Moreover, the approach minimizes latency and computational overhead by limiting LLM invocation to low-confidence cases. Results are shown in Table 4, and demonstrate that the hybrid system outperforms both NERL and LLM alone, achieving an average accuracy of 91%.

Model Configuration	**Average Accuracy**
Gemini 1.5 Flash (fine-tuned)	84%
NERL baseline	85%
Hybrid system (LLM + NERL)	91%

TABLE 4 – Final evaluation of the hybrid system on the posology structuration task.

5 Conclusion

This work introduces MEDPOSOSF , a novel dataset for evaluating the LLMs ability to convert raw textual French posology instructions into a structured format. This task is more challenging than standard NER, as it requires to generate a full JSON output rather than simply tagging spans of text.

To address the limitations of both our NERL baseline and the LLMs on this task, we proposed a hybrid strategy that leverages a surrogate confidence score to combine their outputs, offering a practical and efficient solution for real-world deployment.

Our results also suggest that proprietary LLMs hosted on cloud platforms allow prompt engineering and fine-tuning with less computational resources and less engineering and they perform better than open-source models at comparable levels of latency. However, with more resources than for this work, future work could explore the possibility of using smaller LLMs with domain-specific data, like BioMistral (Labrak *et al.*, 2024), and with guided generation (Willard & Louf, 2023) to improve the performance of open-source models.

Références

ABDIN M., ANEJA J., AWADALLA H., AWADALLAH A., AWAN A. A., BACH N., BAHREE A., BAKHTIARI A., BAO J., BEHL H. *et al.* (2024). Phi-3 technical report : A highly capable language model locally on your phone. *arXiv preprint arXiv :2404.14219*.

ALBERT Q. J., SABLAYROLLES A., MENSCH A., BAMFORD C. & CHAPLOT D. S. (2023). Mistral 7b. *arXiv*.

ALSENTZER E., MURPHY J. R., BOAG W., WENG W.-H., JIN D., NAUMANN T. & MCDERMOTT M. B. A. (2019). Publicly available clinical bert embeddings.

BIANA J., ZHAI W., HUANG X., ZHENG J. & ZHU S. (2024). Vaner : Leveraging large language model for versatile and adaptive biomedical named entity recognition.

BIGEARD E., JOUHET V., MOUGIN F., THIESSARD F. & GRABAR N. (2015). Automatic extraction of numerical values from unstructured data in EHRs. *Studies in Health Technology and Informatics*, **210**, 50–54. Proceedings of MIE 2015, Madrid, Spain.

BROWN T., MANN B., RYDER N., SUBBIAH M., KAPLAN J. D., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A. *et al.* (2020). Language models are few-shot learners. *Advances in neural information processing systems*, **33**, 1877–1901.

CHEN X., OUYANG C., LIU Y. & BU Y. (2020). Improving the named entity recognition of chinese electronic medical records by combining domain dictionary and rules. *International Journal of Environmental Research and Public Health*, **17**(8).

CHIA Y. K., CHEN G., TUAN L. A., PORIA S. & BING L. (2023). Contrastive chain-of-thought prompting. *arXiv preprint arXiv :2311.09277*.

DENG Y., ZHANG W., CHEN Z. & GU Q. (2023). Rephrase and respond : Let large language models ask better questions for themselves. *arXiv preprint arXiv :2311.04205*.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding.

ELHADDAD M. & HAMAM S. (2024). AI-driven clinical decision support systems : An ongoing pursuit of potential. *Cureus*, **16**(4), e57728.

GASCHI F., FONTAINE X., RASTIN P. & TOUSSAINT Y. (2023). Multilingual clinical ner : Translation or cross-lingual transfer ? In *Proceedings of the 5th Clinical Natural Language Processing Workshop* : Association for Computational Linguistics.

GRATTAFIORI A., DUBEY A., JAUHRI A., PANDEY A., KADIAN A., AL-DAHLE A., LETMAN A., MATHUR A., SCHELTEN A., VAUGHAN A. *et al.* (2024). The llama 3 herd of models. *arXiv preprint arXiv :2407.21783*.

GUNASEKAR S., ZHANG Y., ANEJA J., MENDES C. C. T., DEL GIORNO A., GOPI S., JAVAHERIPI M., KAUFFMANN P., DE ROSA G., SAARIKIVI O. *et al.* (2023). Textbooks are all you need. *arXiv preprint arXiv :2306.11644*.

HAAKER T. S., CHOI J. S., NANJO C. J., WARNER P. B., ABU-HANNA A. & KAWAMOTO K. (2025). Approaches for extracting daily dosage from free-text prescription signatures in heart failure with reduced ejection fraction : a comparative study. *JAMIA open*, **8**(1), ooae153.

HE J. & WANG H. (2008). Chinese named entity recognition and word segmentation based on character. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

HE P., GAO J. & CHEN W. (2023). Debertav3 : Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

HOULSBY N., GIURGIU A., JASTRZEBSKI S., MORRONE B., DE LAROUSSILHE Q., GESMUNDO A., ATTARIYAN M. & GELLY S. (2019). Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, p. 2790–2799 : PMLR.

HU Y., CHEN Q., DU J., PENG X., KELOTH V. K., ZUO X., ZHOU Y., LI Z., JIANG X., LU Z., ROBERTS K. & XU H. (2024). Improving large language models for clinical named entity recognition via prompt engineering.

HUANG Z., XU W. & YU K. (2015). Bidirectional lstm-crf models for sequence tagging.

ISARADECH N., RIEDEL A., SIRIKUL W., KREUZTHALER M. & SCHULZ S. (2024). Zero-and few-shot named entity recognition and text expansion in medication prescriptions using chatgpt. *arXiv preprint arXiv :2409.17683*.

JIANG M., CHEN Y., LIU M., ROSENBLOOM S. T., MANI S., DENNY J. C. & XU H. (2011). A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *Journal of the American Medical Informatics Association*, **18**(5), 601–606. DOI : 10.1136/amiajnl-2011-000163.

JOULIN A., GRAVE E., BOJANOWSKI P., DOUZE M., JÉGOU H. & MIKOLOV T. (2016). Fasttext.zip : Compressing text classification models. *arXiv preprint arXiv :1612.03651*.

KRSTEV C., OBRADOVIĆ I., UTVIĆ M. & VITAS D. (2014). A system for named entity recognition based on local grammars. *Journal of Logic and Computation*, **24**(2), 473–489. DOI : 10.1093/logcom/exs079.

LABRAK Y., BAZOGE A., MORIN E., GOURRAUD P.-A., ROUVIER M. & DUFOUR R. (2024). BioMistral : A collection of open-source pretrained large language models for medical domains. In L.-W. KU, A. MARTINS & V. SRIKUMAR, Éds., *Findings of the Association for Computational Linguistics : ACL 2024*, p. 5848–5864, Bangkok, Thailand : Association for Computational Linguistics. DOI : 10.18653/v1/2024.findings-acl.348.

LAHOTI P., GUMMADI K. P. & WEIKUM G. (2021). Detecting and mitigating test-time failure risks via model-agnostic uncertainty learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, p. 1174–1179. DOI : 10.1109/ICDM51629.2021.00141.

LAMPLE G., BALLESTEROS M., SUBRAMANIAN S., KAWAKAMI K. & DYER C. (2016). Neural architectures for named entity recognition. In K. KNIGHT, A. NENKOVA & O. RAMBOW, Éds., *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 260–270, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1030.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2019). Biobert : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240. DOI : 10.1093/bioinformatics/btz682.

LI L., JIN L., JIANG Y. & HUANG D. (2016). Recognizing biomedical named entities based on the sentence vector/twin word embeddings conditioned bidirectional lstm. In *China National Conference on Chinese Computational Linguistics*.

LIANG M. Q., GIDLA V., VERMA A., WEIR D., TAMBLYN R., BUCKERIDGE D. & MOTULSKY A. (2019). Development of a method for extracting structured dose information from free-text electronic prescriptions. *Stud. Health Technol. Inform.*, **264**, 1568–1569.

NAGUIB M., TANNIER X. & NÉVÉOL A. (2024). Few-shot clinical entity recognition in English, French and Spanish : masked language models outperform generative model prompting. In Y. AL-ONAIZAN, M. BANSAL & Y.-N. CHEN, Éds., *Findings of the Association for Computational Linguistics : EMNLP 2024*, p. 6829–6852, Miami, Florida, USA : Association for Computational Linguistics. DOI : 10.18653/v1/2024.findings-emnlp.400.

NEUMANN M., KING D., BELTAGY I. & AMMAR W. (2019). Scispacy : Fast and robust models for biomedical natural language processing. DOI : 10.18653/v1/w19-5034.

PENG N. & DREDZE M. (2017). Improving named entity recognition for chinese social media with word segmentation representation learning.

RENZE M. (2024). The effect of sampling temperature on problem solving in large language models. In Y. AL-ONAIZAN, M. BANSAL & Y.-N. CHEN, Éds., *Findings of the Association for Computational Linguistics : EMNLP 2024*, p. 7346–7356, Miami, Florida, USA : Association for Computational Linguistics. DOI : 10.18653/v1/2024.findings-emnlp.432.

SETTLES B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In N. COLLIER, P. RUCH & A. NAZARENKO, Éds., *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, p. 107–110, Geneva, Switzerland : COLING.

SHEN D., ZHANG J., ZHOU G., SU J. & TAN C.-L. (2003). Effective adaptation of hidden Markov model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, p. 49–56, Sapporo, Japan : Association for Computational Linguistics. DOI : 10.3115/1118958.1118965.

SILVA R. & GOMES L. (2025). An adaptive language model-based intelligent medication assistant for the decision support of antidepressant prescriptions. *Computers in Biology and Medicine*, **190**, 110065.

TEAM G., GEORGIEV P., LEI V. I., BURNELL R., BAI L., GULATI A., TANZER G., VINCENT D., PAN Z., WANG S. *et al.* (2024a). Gemini 1.5 : Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv :2403.05530*.

TEAM G., RIVIERE M., PATHAK S., SESSA P. G., HARDIN C., BHUPATIRAJU S., HUSSENOT L., MESNARD T., SHAHRIARI B., RAMÉ A. *et al.* (2024b). Gemma 2 : Improving open language models at a practical size. *arXiv preprint arXiv :2408.00118*.

UZUNER O., SOLTI I. & CADAG E. (2010). Extracting medication information from clinical text. *Journal of the American Medical Informatics Association*, **17**(5), 514–518. DOI : 10.1136/jamia.2010.003947.

VAN P. P., MINH D. N., NGOC A. D. & THANH H. P. (2024). Rx strategist : Prescription verification using llm agents system. *arXiv preprint arXiv :2409.03440*.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2023). Attention is all you need.

WANG P., QIAN Y., SOONG F. K., HE L. & ZHAO H. (2015). A unified tagging solution : Bidirectional lstm recurrent neural network with word embedding.

WANG S., SUN X., LI X., OUYANG R., WU F., ZHANG T., LI J. & WANG G. (2023). Gpt-ner : Named entity recognition via large language models.

WEI J., WANG X., SCHUURMANS D., BOSMA M., XIA F., CHI E., LE Q. V., ZHOU D. *et al.* (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, **35**, 24824–24837.

WILLARD B. T. & LOUF R. (2023). Efficient guided generation for large language models.

YE Q., AHMED M., PRYZANT R. & KHANI F. (2024). Prompt engineering a prompt engineer. In L.-W. KU, A. MARTINS & V. SRIKUMAR, Éds., *Findings of the Association for Computational Linguistics : ACL 2024*, p. 355–385, Bangkok, Thailand : Association for Computational Linguistics. DOI : 10.18653/v1/2024.findings-acl.21.

A Acknowledgements

We would like to thank Eliott Tourtois for his help in the annotation of the dataset, and in the conception of the structuration schema. We would also like to thank all the colleagues who contributed to the development of the NERL baseline, in particular, Patricio Cerda, Xavier Fontaine, and Baptiste Charnier. We are also grateful for the entire team of Posos for their support and for their indirect contribution to this work. Finally, we would like to thank the reviewers for their valuable comments and suggestions.

B Complete structuration schema

- `as_needed` : dict (optional), contains information about whether the drug must be taken only under certain condition (e.g. if "as needed during headache" is written)
  - `as_needed` : bool, True if the drug must be taken under certain condition
  - `as_needed_for` : str, the specific condition (e.g. "during headache", "if needed", "if pain")
- `designation` : str, the minimal string to which the dosage corresponds, it does not contain the whole instruction, but rather the unit and the type of intake (e.g. "1 comprimé"), this allows to differentiate several posology instructions when needed
- `quantity_and_rate` : dict, contains the amount of intake units to take at once
  - `value` : float, the amount of unit to take at once
  - `unit` : str (optional), the type of unit (e.g. "comprimé(s)", "ampoule(s)", "sachet(s)", "cmp", "cp", "sach", "amp", etc.)
  - `code` : str (optional), the SNOMED code for the unit
- `max_dose_per_period` : dict (optional), contains information about a potential maximum dosing
  - `dose` : int, the maximum amount of units
  - `dose_unit` : str (optional), the type of unit
  - `code` : str (optional), the SNOMED code of the dose_unit
- `timing` : dict, contains the timing with which each intake must take place
  - `bounds_duration` : dict (optional), the duration during which the treatment occurs
    - `max_value` : float (optional), the maximum amount of time units the treatment must last
    - `value` : float, the amount of time units the treatment must last (if max_value is not null, value reperesent the minimal value)
    - `unit` : str, the unit for measuring the treatment duration ("hours", "day", "week", or "month")
  - `bounds_period` : dict (optional), an alternative to bounds_duration where exact dates have been provided
    - `start_date` : str, start date in format YYYY/MM/DD
    - `end_date` : str, end date in format YYYY/MM/DD
  - `day_of_week` : list (optional), list of weekday diminutives when the treatment must be taken (e.g. "mon" or "sat")
  - `frequency` : int, an integer for the number of intake for the given period (e.g. the 2 in "twice a day")

— `frequency_max` : int (optional), an integer for the maximum amount fo intake for the given period (frequency is then the minimum)
— `number_repeats_allows` : int (optional), the number of times the treatment can be renewed
— `offset` : str (optional), substring extracted from the query that indicates what is the offset of the intake ("30 minutes" in "30 minutes before meals")
— `period` : int, the number of period_unit to observe before repeating the intake (6 in "3 tablets every 6 hours")
— `period_unit` : str, the time unit used for measuring the period between intakes ("hours" in "3 tablets every 6 hours")
— `sequence` : int (optional), when several posology instructions are given, sequence indicates their order
— `time_of_day` : list (optional), indicates precise intake hours in the format HH :mm :ss. `time_of_day` and `when` cannot be simultanously non-empty
— `when` : list (optional), indicates the moment of intake relative to the patient activity (e.g. "AC" for before a meal), the list of possible values is given in the FHIR standard [15]

C Dataset statistics

Statistics	
Number of queries	129
Number of posology instructions	131
Minimum number of tokens	2
Median number of tokens	9
Maximum number of tokens	29
Minimum number of occurrences of a JSON field	1
Median number of occurrences of a JSON field	69
Maximum number of occurrences of a JSON field	131

TABLE 5 – Statistics of the MEDPOSOSF dataset.

D NER training dataset

E Prompt details

1. Chain of thought examples

here is an exmaple 1.1 how to proceed.

sentence : `ampoute , 1 fois par trimestre besoin`

step by step reasonning :

15. https://hl7.org/fhir/R4/valueset-event-timing.html

Entity types	Number	Example
BOUNDS	647	pendant 7 jours
DOSE	1,188	2cp
DRUG	1,007	EFFICORT Hydrophile
FORM	574	CPR
FREQUENCY	213	3 fois
NUMBER_REPEATS	20	1 Boîte.a renouveler
PERIOD	555	par jour
REASON	143	SI BESOIN
STRENGTH	721	200 mg
TIME_OF_DAY	19	à 08h
WHEN	723	le matin

TABLE 6 – NER entities present in the training dataset

0. Check spelling, remove punctuation and make the sentence more explicit : `ampoute , 1 fois par trimestre besoin` most likely means `1 ampoule 1 fois par trimestre si besoin`

1. Check whether there are several entities. Here, ampoute (misspelled) refers to only one entity.

2. I see that the sentence refers to medicines, so I can associate `category` with `MEDICATION`.

3. Since there is only one entity, the `designation` can be associated with the value of the entire sentence : `ampoute 1 fois par trimestre besoin`

4. We can see that the sentence talks about ampoute (misspelled), 1 time per trimester, so it's an `entity_type` to be associated with the `QUANTITY` value.

5. The `quantity_and_rate` will contain the values `type` associated with `DOSE` (one ampoule corresponds to one dose of product), unit associated with `ampoule(s)` and `value` associated with `1` (because once per quarter in the sentence). There are no elements relating to the maximum value, so there's no need to set `max_value`.

6. The `timing` section. The value of `bounds_duration_text` will be equal to `1 fois par trimestre`, which means that the `frequency` will be equal to `1`, the `period` to `3` and the `period_unit` to `month` because a quarter is equal to 3 months.

7. Concerning the `as_needed` section, as it writes `besoin` which probably means `si besoin`, we'll set the `as_needed` value to `True` and the `as_needed_for` value to `besoin`.

The result is the following YAML format :

```
entities:
- a_needed:
    as_needed: true
    as_needed_for: besoin
  category: MEDICATION
  designation: ampoute 1 fois par trimestre besoin
  entity_type: QUANTITY
  quantity_and_rate:
```

```
      type: DOSE
      unit: ampoule(s)
      value: 1
    timing:
      bounds_duration_text: ''
      frequency: 1
      frequency_texts:
      - 1 fois par trimestre
      period: 3
      period_unit: month
  posology_string: ampoute 1 fois par trimestre besoin
```

2. Few-shot examples

here is an exmaple 2.1 how to proceed.

sentence : `0.5 à 1 cp au coucher si besoin (1 boite AR)`

```
  entities:
  - category: MEDICATION
    designation: 0.5 a 1 cp au coucher si besoin
    as_needed:
      as_needed: true
      as_needed_for: si besoin
    entity_type: QUANTITY
    quantity_and_rate:
      max_value: 1.0
      type: DOSE
      unit: comprimé(s)
      value: 0.5
    timing:
      bounds_duration_text: ''
      frequency: 1
      frequency_texts:
      - au coucher
      period: 1
      period_unit: day
      when:
      - HS
  posology_string: 0.5 à 1 cp au coucher si besoin
```

3. Mistakes to avoid

If a time of day is specified in hours, don't forget to specify it in the `time_of_day` field (even if the when field is also filled)

Be careful with time of intake : `le matin 30 minutes avant le repas` is not two different time of day for in the morning and before meal (`MORN`, `AC`) but it represents a single one for

taking the medication before the meal of the morning (ACM)

For time of days that fit in the when field, do not mistake meals (like ACM for breakfast and CD for lunch) with periods of the day (like MORN for morning and NOON for mid-day), you must use a meal time only if the meal is explicitly described

Be careful with as_needed field, it should include as_needed equals True and as_needed_for referencing condition, if you detect relevant condition in text for the entity in question (si besoin, si migraine, si angoisse). as_needed field should not constitute a separate entity.