Assessing Large Language Models for Structuring Patient Records

Eduardo M. Campos[1], **Rafael C. G. Conrado**[1]
Caetano Traina Jr.[2], **Agma J. M. Traina**[2], **Mirela T. Cazzolato**[2]

[1]Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP)
Universidade de São Paulo (USP) – Ribeirão Preto, SP – Brazil

[2]Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP) – São Carlos, SP – Brazil

{educampos,rafaelconrado}@usp.br, {caetano,agma,mirela}@icmc.usp.br

Abstract. *Patient Records are digital documentation of a person's presence in a health institution. This data has been used to enhance medical experience and has allowed for many breakthroughs in scientific research. Nevertheless, the unstructured and non-standard nature of medical notes constitutes an obstacle to the complete usability of this information in crucial tools such as data analysis, visualization, and machine learning algorithms. In this work, we evaluated the use of Large Language Models to preprocess the unlabeled data into a defined structure, test different model sizes and architectures, and develop prompt strategies and fine-tuning techniques. The proposal was evaluated using the Covid-19 DataSharing dataset. Even low-computation cost models can achieve great performance, which could enable an approach to standardize many kinds of data.*

1. Introduction

Patient records, often maintained as Electronic Health Records (EHRs), are historical notes that represent diverse information of a person's clinical life and are currently a valuable asset in decision-making and health care. This format has been quickly adopted in the current trend of big data, *e.g.*, in the United States, it jumped from 10% to 96% in 10 years [Jiang et al. 2023] and in Brazil, over 85% of hospitals had an electronic system to track patients' conditions [Barbalho et al. 2022].

EHRs' contain information about single exams, drugs taken, patient condition, personal information, temporal evolution, etc. EHRs constitute a wide terrain of sub-explored raw data that has assisted in understanding diseases and proposing treatments. However, this abundance of information that makes EHRs interesting comes along with an equally vast amount of ways to represent it, which imposes a barrier for traditional machine learning methods that rely on repetitive patterns and can easily be misled by note inconsistencies, multimodal representation, grammar errors, and field-specific terms.

To overcome this issue, we evaluated the use of LLMs, which are an algorithm capable of mimicking human-like responses in generative answers and accomplishing good results in many tasks, such as text categorization, to structure patient records [Costa et al. 2024]. The most powerful and best-performing models require a huge amount of data and computer power to be trained on. Given the heterogeneous access to resources, this necessity could manifest as a hindrance for many institutions. Accordingly, we focus on small to middle-sized pre-trained models that are open-source, can execute on personal computers, and can get reasonable outputs with no specific fine-tuning.

2. Background

In this section, we present the relevant background.

Large Language Models(LLMs). The desire to have computer systems to understand and process human language initiated alongside computation itself, with the early models (1950s-1950s) relying on Chomsky's theory of grammar to perform basic language translation [Hadi et al. 2023]. However, these models, as well as subsequent statistical language models, had struggled to comprehend the complex structure of semantics and context [Hadi et al. 2023].

In the 2010s, deep learning in the form of recurrent neural network language model (RNNLM) made an advancement in the area, generating more human-like responses and better at capturing nuances in writing, but had its performance limited due to its inability to recognize longer-term dependencies in language as well as not allowing parallel computing[Vaswani et al. 2017].

These problems were overcome in 2017 [Vaswani et al. 2017] with the transformer mechanism that enables the creation of models with a huge amount of parameters GPT-1 (2018), LLama (2023), Deepseek (2024) that could do several NLP tasks such as text summarization, sentiment analysis, machine translation, text generation with human-like responses that were coherent.

The first LLM models followed the previous encoder-decoder architecture in which an input is converted into a set of tokens in an N-dimensional space representing the meaning and the position of the given sentence. This vector will be passed through a series of Multi-Head Attention layers in the encoder that perform Dot-Product Attention. The idea behind the structure is that each head will extract different parts of the complex relation between words in the parameter training and retrieve it afterward. Subsequently, the updated vectors undergo an MLP that helps to map the input and passes it to the encoder. The encoder follows a similar Multi-Head Attention mechanism but has a masking feature that hides part of the sentence, as this part is responsible for the next token generation.

Prompt Engineering. Large Language Models, being billions of parameter machines that have to be carefully fine-tuned through an enormous amount of data and computer power, may cause the impression that all of their capacity relies on the parameter training itself. However, the input given to the model is a crucial piece in determining its performance.

Prompt engineering studies strategies to enhance an LLM's ability to perform a specific task by structuring a prompt in a certain way or passing additional information with it [Liu et al. 2023]. The most common techniques are Zero-Shot Prompting, in which a description of how to do the task is used to help the factual knowledge learned previously; Few-Shot Prompting, which gives the model some examples of how to do the desired assignment in addition to the description [Brown et al. 2020]; and Chain-of-

Thought (CoT) Prompting that ask them with the prompt for a step-by-step reasoning process in the response [Wei et al. 2022].

Temperature tuning. Many strategies have been invented to increase LLMs' accuracy and avoid hallucination in a specific task. Prompt engineering techniques focus on different ways of passing a prompt to a model. Another aspect of models that can be adjusted is their hyperparameters, which are characteristics that modify certain behaviors and limitations in an LLM. Across the hyperparameters, there is the top-k sampling that restricts the next token to be output among the k most likely ones, and a repetition penalty that punishes the model for repetitive patterns in the response.

Temperature is a parameter that affects the token chosen in an iteration. After passing through the multi-head attention mechanism, a softmax function is applied to assign a probability for the next token. The temperature defines how greedy the model will be, always choosing the better-ranked token or allowing the model to create alternative paths. This parameter is often appointed as a creativity filter.

3. Methodology

In this work, we aim to measure how well an LLM could understand the data present in an EHR, and, along with previously known knowledge, convert the instance into a different structure. The models we choose are all Small- to Medium-sized pre-trained parameters. This setting is important to avoid using open APIs for security and privacy reasons that come with sensitive data in medical records and for scalability purposes since medical institutions have heterogeneity in computer resources available. Table 1 lists the selected models.

Table 1. Models employed in the study, with the creator, license, and number of parameters.

Model	Creator	license	Parameter size
Gemma	Google	open-source	7b
Gemma3	Google	open-source	12b
Gemma2	Google	open-source	9b
Deepseek-r1	Hangzhou	open-source	7b
Llama3	Meta	open-source	8b

Database. We used the Fapesp Covid-19 DataSharing records [FAPESP 2020], a Brazilian initiative to share data collected in many hospitals during the pandemic. The dataset has EHRs collected from hospitals of São Paulo state, including demographic information, exams and analytes, and outcomes.

Proposed evaluation: Strucute conversion test. The experiment to measure the ability of LLMs to struct data carried in this study aimed to evaluate if the models could determine a label $L \subseteq \{\text{reagent, Non-reagent}\}$ given a numeric result for an exam and the reference value. This assigned label would be costly for a human since those results and reference values were recorded in many different ways, some with grammar errors, some with pure numbers, and some with a mix of textual and numbers. The intent to automate a program for every exam, given its heterogeneity, would be as hard as to analyze it by hand.

Figure 1 shows the proposed evaluation pipeline. In the beginning, all EHRs are converted to tables in a Relational Dataset.

The user accesses the database and selects the desired exam data to be analized. The original data is organized in attributes, but its content is mostly unstructured (e.g., textual and hand-free notes). Based on predefined prompts, LLM models structure the information related to the exam reference, values, and result. For this task, the unstructured data feeds the LLM model, which composes the final prompt to structure the exam. The resulting information is stored in the database for future use.

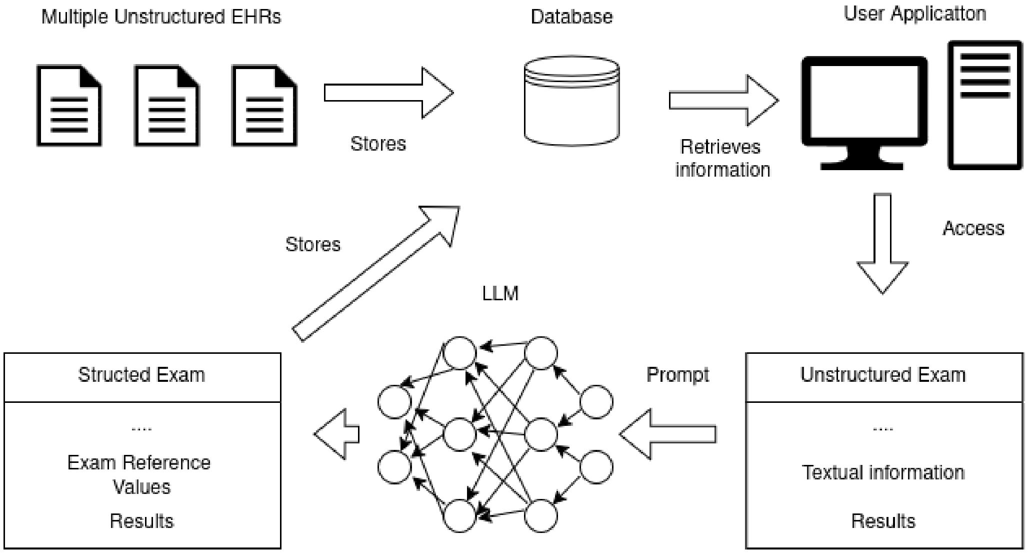

Figure 1. The proposed methodology: We explored a set of EHRs containing textual information that stored in a Relational Database. The user retrieves unstructured patient data, and runs predefined prompts in the LLM models to generate structured information related to exam reference, values and results.

Table 2 shows an example of prompt employed to convert an exam. The prompt is in Portuguese, the same language of the original data. The output is a category among ‘reagente’, ‘não reagente’, or ‘indeterminado’, that indicates the exam result based on the input value and the reference information. All other prompts developed in our methodology is avaliable at a GitHub repository (see Section 6).

4. Experiments

Models’ Accuracy. Figure 2 shows the models’ accuracy when using different prompt strategies. We can see that the change in performance is relevant for all models except Deepseek and Gemma3. The first one had a huge language and format obstacle, which could have drained any prompt advantages. The second one had almost 100% accuracy with a Zero Shot prompt. Gemma1 and Llama3 followed the expected pattern with more complex prompts leading to better responses, and Gemma2 had an unexpected greater result using ZeroShot.

Table 2. Example of prompt convert an exam analyte result into a categorical result between 'reagente', 'não reagente', or 'indeterminado'.

```
'Zero-shot' Prompt: analyte result into a categorical result

Contexto: voce é um médico e tem que determinar se o
seguinte exame é reagente ou não_reagente:
Exame: analito: {DE_ANALITO1},
resultado: {DE_RESULTADO1},
valor_referencia: {DE_VALOR_REFERENCIA1}
formatação: formate sua resposta da seguinte maneira:
resposta:[reagente ou não_reagente ou Indeterminado]
```

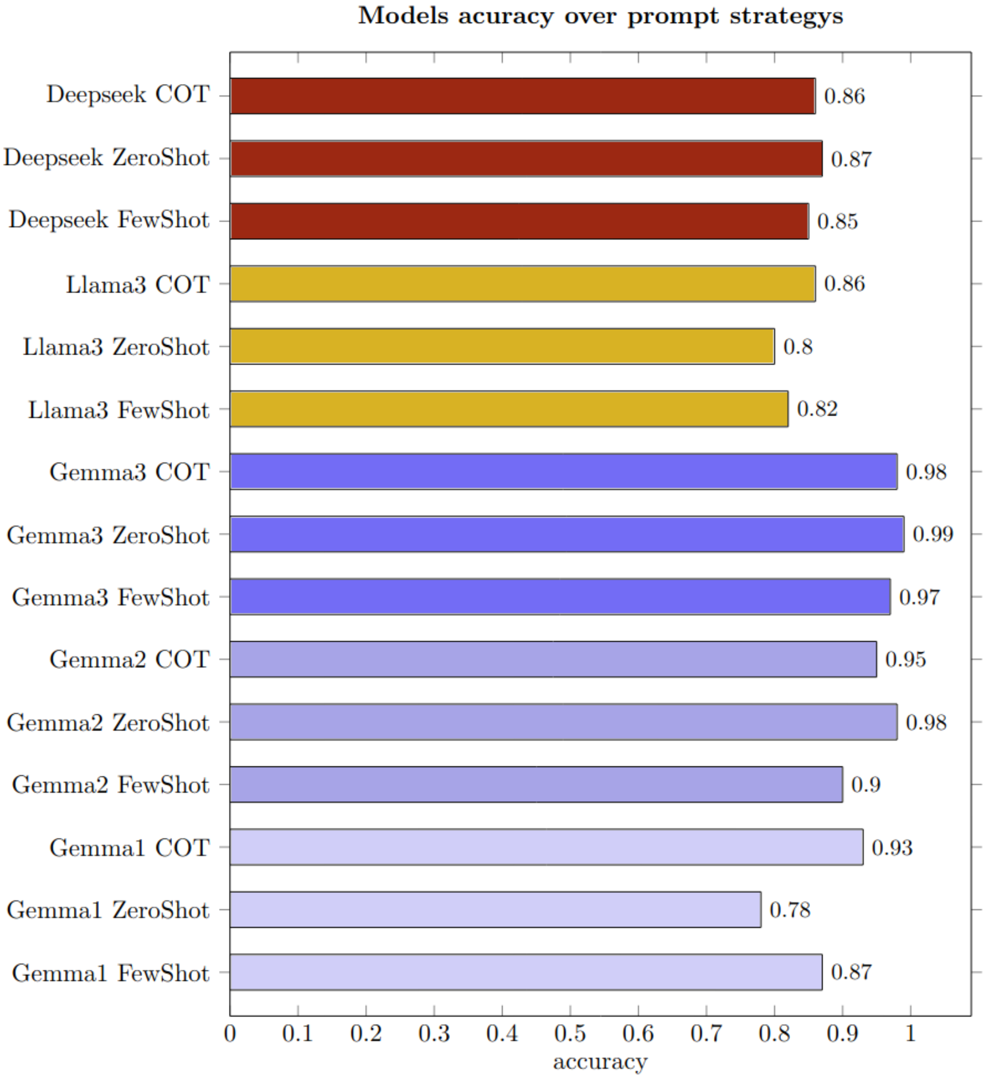

Figure 2. Models acuracy over different prompt strategies: Gemma3 FewShot and Gemma2 ZeroShot presented the best results.

Temperature Setting. In this experiment, we evaluated the performance selected models over the distinct values of temperature. Figure 3 shows the Accuracy results. We can see the models performed slightly better at low temperatures, what intuitively is expected for a noncreative (object) task. Also, the models had a linear decrease in accuracy until a plateau around 0.5 of temperature, but overall the accuracy variation was small. Gemma3 presented the best results for all values of temperature, followed by Gemma2.

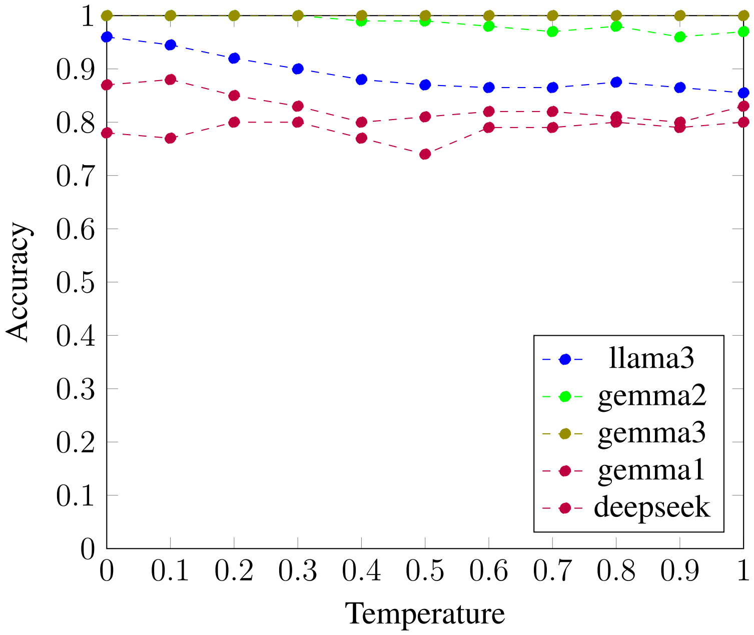

Figure 3. Accuracy of evaluated models according to the temperature value.

5. Limitations

Although it appears that all models performed well at the chosen task, we cannot extrapolate those results to more complex tasks or other datasets without prior testing. Furthermore, clinical data has a sensitive nature that prohibits even tiny errors while handling it.

6. Conclusions

In this work, we evaluated the the use of LLMs to struct pacient redords.this data is rich in useful information that guides Medical facilites.however, it's stored in many different ways, over institution or even over exams,what restricts its full potencial use. We measure how well small- to medium-sized models comprehend the concepts given in an exam and apply changes to their form so that a general form can be achieved.

In our work all the models showed a good performance. Independent of the the prompt strategy of hyperparameter tunning.what enable a cost efficient and general why to preprocessing this records.finally to expand this results it's needed to understand if the great performance would stay still in more complex exams or different institutions.

Our code and complete prompts are open-sourced at GitHub[1].

[1] GitHub repository: https://github.com/Dudu-Campos/Llms-EHR

Acknowledgments

We thank the support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001, the São Paulo Research Foundation (FAPESP) – grants #2016/17078-0, #2023/14390-7, #2024/13328-9, the Brazilian National Council for Scientific and Technological Development (CNPq), PIBIC-CNPq – grant #125355/2024-6, Programa Unificado de Bolsas de Estudo para Apoio e Formação de Estudantes de Graduação (PUB-USP) – grant #2075, and the University of São Paulo (PRPI Ordinance No. 1032) – grant #207.

References

Barbalho, I. M. P., Fernandes, F., Barros, D. M. S., Paiva, J. C., Henriques, J., Morais, A. H. F., Coutinho, K. D., Coelho Neto, G. C., Chioro, A., and Valentim, R. A. M. (2022). Electronic health records in brazil: Prospects and technological challenges. *Frontiers in Public Health*, 10. DOI: http://dx.doi.org/10.3389/fpubh.2022.963841.

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. DOI: http://dx.doi.org/10.5555/3495724.3495883.

Costa, L., Figênio, M., Santanchè, A., and Gomes-Jr, L. (2024). Llm-mri python module: a brain scanner for llms. In *Anais Estendidos do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 125–130, Porto Alegre, RS, Brasil. SBC. DOI: http://dx.doi.org/10.5753/sbbd_estendido.2024.243136.

FAPESP (2020). FAPESP COVID-19 Data Sharing/BR. Technical report, FAPESP. URL: https://repositoriodatasharingfapesp.uspdigital.usp.br.

Hadi, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Wu, J., Mirjalili, S., et al. (2023). A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Jiang, J., Qi, K., Bai, G., and Schulman, K. (2023). Pre-pandemic assessment: a decade of progress in electronic health record adoption among us hospitals. *Health Affairs Scholar*, 1(5):qxad056.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35. DOI: https://doi.org/10.1145/3560815.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. URL: proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

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.