Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Beyond ChatGMP: Improving LLM generation through user preferences

Fiammetta Caccavale[a], Carina L. Gargalo[a], Krist V. Gernaey[a], Ulrich Krühne[a*], and Alessandra Russo[b]

[a] Process and Systems Engineering Center (PROSYS), Department of Chemical and Biochemical Engineering, Technical University of Denmark, Søltofts Plads, Building 228 A, 2800 Kgs. Lyngby, Denmark
[b] Department of Computing, Imperial College London, London, UK
* Corresponding Author: ulkr@kt.dtu.dk.

ABSTRACT

Prompt engineering – improving the command given to a large language model (LLM) – is becoming increasingly useful in order to maximize the performance of the model and therefore the quality of the output. However, in certain instances, the user is not able to enrich the prompt with additional and personalized details, such as the preferred tone and length of generated response. Therefore, it is useful to create models that learn these preferences and implement them directly in the prompt. Current state-of-the-art inductive logic programming (ILP) systems can play an important role in the development and advancement of digitalization strategies. For example, they can be used to learn personal preferences of users without sacrificing human interpretability of the learned outcomes. These systems have recently witnessed the development of data efficient, robust, and human interpretable algorithms and systems for learning predictive models from data and background knowledge. In this paper, one of these systems, ILASP (inductive learning of answer set programs), is used to develop a proof of concept of how personal preferences of groups of students participating in an interview exercise can be learned to tailor and improve the generated response of a LLM used in an educational context.

Keywords: Intelligent Systems, Machine Learning, Artificial Intelligence, Education, Industry 4.0.

INTRODUCTION

One of the primary issues concerning the application of artificial intelligence (AI) in education is the reliability and transparency of the information it generates. In particular, widely accessible generative AI models such as large language models (LLM) are designed to deliver responses even when the correctness of those responses might be questionable. A known phenomenon currently occurring with available chatbots is hallucination, meaning that the model provides incorrect or misleading information. To mitigate this issue, a possible solution could be to reduce the generative abilities of these models by opting for more conservative hyperparameter settings. Among these there is temperature, which controls the randomness of the output. Higher values produce more creative results, while low temperature produces more deterministic and conservative outputs. Other hyperparameters that control the randomness and length of the generated answer are top p, token length and max tokens. Top p computes the cumulative probability distribution, setting a cut off when the distribution exceeds the value. For example, a top p value of 0.3 means that only the tokens comprising the top 30% probability mass are considered. Researchers can use all these hyperparameters to tune how creative or long a response should be, having more control over the generative process. Another strategy for increasing the control over the generation of the LLM is prompt engineering. In this case, the AI model could be combined with more interpretable techniques to enhance the prompts given to the LLM.

The Department of Chemical and Biochemical Engineering at the Technical University of Denmark (DTU) offers a course in Good Manufacturing Practices (GMP). As part of the course, students are required to participate in an audit exercise, in which they take on the role of an auditor investigating a company about its good manufacturing practices. In the audit exercise, the company is fictional and is represented by the teachers. After a few

years of participating in this exercise, teachers have agreed to test the replacement of the physical teacher with ChatGMP [1], an AI-powered digital audit tool to represent the fictional company. The chatbot, tested at DTU in the spring of 2024, proved to be considered a viable alternative by both teachers and students. To be noted is that part of the role of a teacher in educating students is also to understand the personal differences of each student and leverage them to help students achieve the best learning outcome. For example, in the audit exercises the teachers normally deduce whether students prefer to have an elaborate or straight-to-the-point answer and try to act accordingly. Moreover, they would perhaps try to help students in case they realized that they do not understand the answer provided, for example, by repeating it but in a simplified manner. Additionally, teachers would also detect whether students missed asking very important questions for the positive outcome of the exercise. In fact, not asking some important questions could result, in this exercise, in the students not obtaining the correct documents that they need to write an audit report. However, most commonly available chatbots would reply to each user and each question in the standard format, without taking these differences and learned knowledge about the students into account.

An approach that can automatically extract knowledge from given examples and therefore could be used to address both the interpretability and the lack of personalization challenges expressed above is inductive logic programming (ILP). The field of ILP has been subjected to substantial advancements over the course of the past ten years. These systems are capable of automatically extracting knowledge (expressed as logic program) from given examples and domain-specific prior knowledge. The aim of ILP systems is to learn models that together with existing background knowledge, explain labeled data [2-4]. An advantage of these models is that they are easily expressed in plain English and are inherently interpretable. This means that they can be explained to a human user, which in turn can facilitate a closer interaction between humans and machine.

This work aims to investigate the use of ILP to learn user preferences and provide a proof of concept on how these can be injected into the prompt given to ChatGMP. To achieve this, we investigate three main use cases: (1) detect whether students have missed some important question in their audit preparation work, (2) analyze if there are some repetitive questions that the students are asking, and (3) extract user preferences to personalize the response generated by the LLM. Overall, the aim of these investigations is to extract insights and improve the question-answering process. This will be done by (i) providing valuable feedback to the students, (ii) further improving the model's responses, and (iii) learning user preferences to tailor the audit exercise to each individual group of students.

BACKGROUND

Current state-of-the-art ILP systems such as ILASP (inductive learning of answer set programs) [2] and FastLAS [5] can learn a broad range of interpretable models using answer set programming (ASP) [6]. ASP is a purely declarative language for representing and reasoning about knowledge. Typically, real-world problems are addressed by manually encoding them into ASP programs, followed by computing answer sets, which serve as solutions to the original issues. Due to its expressiveness and ability to efficiently find solutions, ASP is gaining increasing interest in the industry [7].

Cutting-edge ILP systems, to automatically derive models from labeled data, utilize input answer sets that represent positive or negative solution examples of a real-world problem and, alongside existing knowledge, learn ASP programs that encode the original problem. ILP has been effectively utilized in numerous real-world applications, including bioinformatics and chemoinformatics [8-10], natural language processing [11], and web mining [12]. Specific cases within chemistry and engineering include interactive exploration of chemical fragments [13], uncovering hydrogen-bonding patterns in crystals [9], drug design [14] and mining pump control systems. These methodologies highlight the capabilities of ILP in different fields and applications, including education.

MATERIALS AND METHODS

This work aims to take the experiment performed with the creation of ChatGMP a step further. In fact, it aims to present a proof of concept on how the responses generated by ChatGMP could be improved through the use of ILP.

Data

To leverage a pre-trained LLM on a specific task, domain specific data needs to be collected and curated.

The data used in this work is mainly collected from the audit exercises performed during the GMP course; the audits were performed and video recorded in the spring semesters of 2022 and 2023, summing up to 35 audits of approximately 45 minutes length each. The audios were then extracted from the videos in Python. To detect the different speakers and therefore being able to identify the questions (asked by the students) and the answers (formulated by the teacher), diarization, a process of automatically partitioning an audio recording into segments that correspond to different speakers, was performed. The speaker segmentation was then used to extract the text through a speech-to-text model. Then all

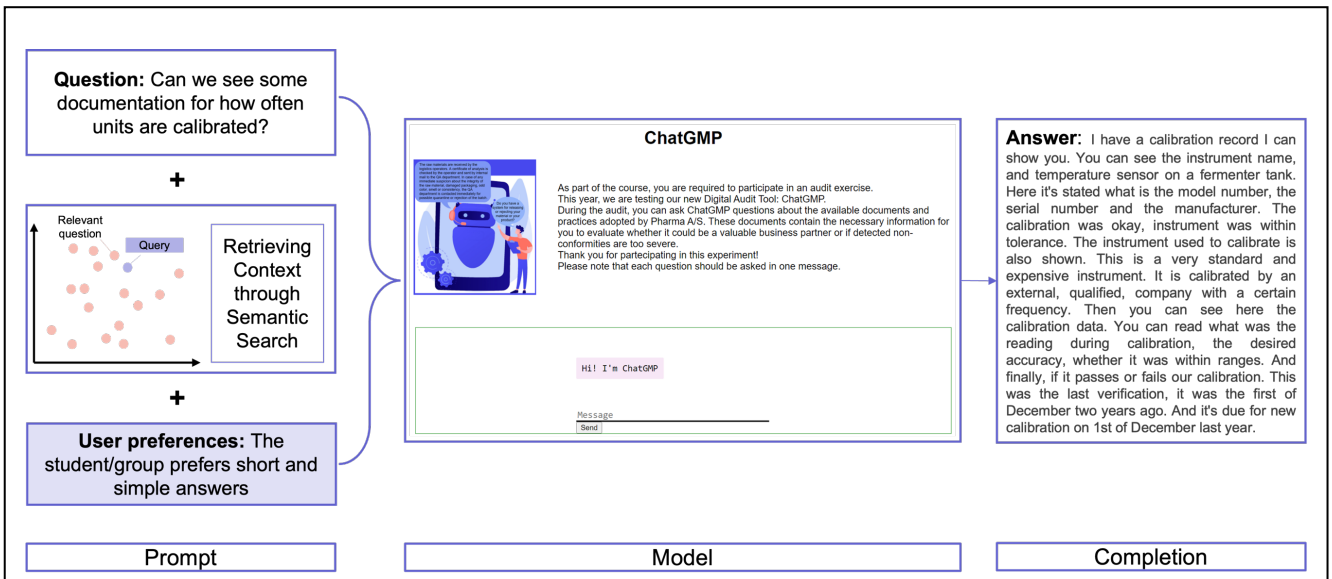

Figure 1: Schematics of the prompt given to ChatGMP. ChatGMP is prompted to open the "Calibration record" document, and summarizes it to the students. Here, the prompt is enriched with user preferences, which are weak constraints learned in ILASP. Figure is adapted from [1].

the students' questions were merged, therefore resulting in a document with the question-answer pairs. To ensure the highest quality of data to be used as model input, the resulting text documents (one for each audit performed), were manually checked and edited when necessary. Finally, the curated documents containing the question-answer pairs were concatenated and saved in a JSON file, for easiness of use. Further details can be found in [1].

ChatGMP Model

ChatGMP is a question-answering (QA) LLM based on a pre-trained model and delivered to the students through an interactive user interface. The LLM component is based on a pre-trained model, FLAN-T5 base [15].

The prompt includes the question asked by the student group performing the audit and is enriched with some context, which contains the response to the question. The context is retrieved through semantic search by calculating the cosine similarity between the query question and the most similar question in the corpus. The LLM then generates an answer to the question asked based on the context provided. Further details are discussed in [1].

ILP tasks and objectives

For the logic-based part, ILASP is used to learn rules concerning the cases investigated (missing and repetitive questions) and weak constraints to understand the preferences of each group of students. These preferences might concern, for example, the length or level of detail of the questions that they ask, the topic that they focus on more, whether they repeat questions or they are missing fundamental questions to achieve a high grade. The objective of this project is to automatically feed this information and therefore enrich the prompt given to ChatGMP with the weak constraints and user preferences learned. In this manner, ChatGMP could personalize the response based on the user. This process is shown in Figure 4.

The three different cases investigated are:

- *Missing questions*: To achieve a good performance in the audit report, there are two very specific questions (i.e. reference questions) that should be asked. We vectorize both the reference questions and the asked questions and we calculate the cosine similarity between them. Then, based on the similarity between the reference questions and the asked questions, we learn rules to establish whether the group of students asked all questions ("correct" label), if they asked one of them ("semi-correct" label) or if they did not ask any of the reference questions ("wrong" label). The aim of this use case is to directly assess the quality of the questions prepared by the students by checking whether some important questions have not been asked by the group of students, inevitably resulting in a lower grade. Students will then receive feedback on their list of questions, spurring them to further reflection.
- *Repetitive questions*: In the audit exercise, students might not understand a concept and ask for further details or repeat the same question more than once. We assume that question

repetition means that the answer given by the teacher/ChatGMP was inadequate, being either incomplete or not understood by the students. Therefore, we want to predict if students will ask questions with high similarity (repeating one question) to evaluate the understanding of the students. The aim of this use case is to gain deeper insight into the understanding of the students and adjust the agent's answer accordingly, by providing additional details or improving on the quality of the future generated answers.

- *User preferences*: In this case, we want to learn both general rules to achieve a good performance (e.g. questions about a specific topic are to be preferred to another one to maximize performance), as well as personal preferences of the groups, such as length and complexity of the questions, order of the topics, in order to better engineer the prompt given to the LLM for the answer generation. The aim of this use case is to learn user preferences to personalize the responses provided by ChatGMP according to each groups' individual traits and favorable factors.

The examples provided to ILASP were automatically generated using Python and extracted from historical audits recorded in the spring of 2022 and 2023. Explaining the formalism behind ASP is out of the scope of this work. However, from the provided code and learned rules it is possible to extract and discuss some general patterns and insights on the audits.

RESULTS AND DISCUSSION

The goal of this study is to improve the prompt given to ChatGMP with user preferences learned in ILASP. These rules are easily translatable to natural language and therefore can be appended to the prompt.

Regarding the first case investigated, whether the students are missing important questions, ILASP founds the following rule:

```
relevant(V1) :- similarity(V1,V2); V2 > 62.
```

Provided the given examples and the label for the three classes ("correct", "semi-correct" and "wrong"), ILASP was able to automatically calculate the threshold for differentiating the classes. V1 indicates the reference question (either D1 or D2, since we have two relevant questions that the students should ask). V2 is the cosine similarity between V1 and the asked question. This means that if the cosine similarity between the asked question and the reference question is less than 62, the reference question was not asked. When the example is "correct", it means that the group managed to ask both the reference questions needed. "Semi-correct" means that one of the two questions was asked, while "wrong" indicates that the students did not ask any of the reference questions.

For detecting repetitive questions, the following rule was learned:

```
repeat :- success(V1,V2); repetition(V1,V2,V3); V3 > 60.
```

In this example, V1 and V2 indicate the question sequence (e.g., question 1 and question 2), and V3 is the cosine similarity between the two. Therefore, if two questions in have high cosine similarity, it means that the question is repeated. Here, the threshold learned is 60, so if the cosine similarity of two consequent questions is higher than 60, it indicated that students might be asking the same questions again, maybe because they did not understand the first answer provided by the chatbot.

This rule is useful to extract because, by calculating this similarity of the questions asked by the students, we can learn whether students did not understand an answer given by ChatGMP, so asked a similar question, and inject this information in the prompt of the LLM.

In this case, a possible prompt could be:

- "Please answer the <QUESTION> based on the given <CONTEXT> in a more elaborate manner, since students seem not to have understood the previous response".

This would ensure a more elaborate, complete and perhaps simplified answer from ChatGMP. On the other hand, it is also possible that the chatbot did not understand the question in the first place and therefore prompting it to re-elaborate the answer might lead to a better response. This approach would hopefully lead the students to understand the response and get the answer they were inquiring.

These two use cases are particularly important because the knowledge extracted by ILASP can be helpful when building a chatbot. In fact, detecting whether some questions are missing or if the students did not understand something would enable us to provide feedback back to the students and help them. This would thus ensure that the students have understood and learned the necessary material. This is of the utmost importance, since students' learning should be the ultimate goal. Thus, AI should not be blindly used to replace tasks to save time and resources but rather to help the educator with the automation of repetitive tasks while still ensuring that students reach the learning objectives.

In the last case, concerning user preferences, these are the rules learned:

```
:~ question_type(V1, review), number_questions(V1,
V2).[-V2@3, V1, V2]
 :~ question_type(V1, logistics), number_ques-
tions(V1, V2).[V2@1, V1, V2]
```

This investigation aims to learn the personal preferences of students, for example by learning the preferred number of questions (if many or few) and the length of the outcome. It does it by learning weak constraints. In the learned rule, V1 is the topic of the question, which is defined among 7 topics related to GMP, which are: management, training, quality assurance, quality control, maintenance, production, review and logistics. ILASP learned that the topic is the most important indicator for performance, followed by the number of questions that were asked about the topic. Additionally, how complex the questions are, and the level of details are also relevant considerations.

According to the learned preferences of each user, a possible prompt could be engineered by adding the following request:

- "Please answer the <QUESTION> based on the given <CONTEXT> *in a short and precise manner*".
- "Please answer the <QUESTION> based on the given <CONTEXT> *to include more information*".
- "Please answer the <QUESTION> based on the given <CONTEXT> *and recommend the students to ask more about maintenance*".

Using the weak constraints learned, every prompt could be tailored for the individual student or group of students interacting with the chatbot. This approach also enables to extract which topics, historically, lead to the best results (e.g., grades in the report). This means that, for example, if students ask more questions about logistics, they will get higher grades in the audit report. So, ChatGMP could also use this information to provide feedback to students on some topics they are missing or did not gather enough information about.

Although this study only provides a proof of concept and the ILP approach was not yet integrated into ChatGMP, it aims to give an example of how the prompt of an LLM could be improved through interpretable and reliable insights, incorporating use preferences and tailoring the prompt to the individual user.

CONCLUSION

AI chatbots are increasingly being used in various contexts, including in education. It is necessary to ensure ethical applications of these models in order to safeguard students, as well as provide the correct and relevant information. Additionally, every student is different, and therefore the responses provided by an LLM should be tailored accordingly, to enhance the learning experience and leverage this individuality. This study provides a proof of concept on how ILASP can be used to learn rules and weak constraints in an audit exercise scenario. This results in the ability to enrich the prompt given to the LLM by including feedback, such as missing questions and lack of understanding, and user preferences. This enables the prompt to be tailored according to the individual user interaction, personalizing the responses and providing instant feedback to students.

FUTURE PERSPECTIVES

In the near future the second (repetitive questions) and third (user preferences) use case will be combined and further developed to enrich the prompt of ChatGMP. This will ensure that the LLM will receive further information on how to answer the questions. Moreover, by learning the user preferences, the length, complexity and level of details contained in the generated answer can be calibrated based on individual preferences. Once the investigated cases will be integrated into the pipeline of ChatGMP, the role of the chatbot will therefore be to understand how the interaction is evolving, so that it can adapt its answers to improve the learning and aid students further. This is a dynamic approach, since the interaction will be observed in real time rather than a posteriori.

DIGITAL SUPPLEMENTARY MATERIAL

The ILASP code used is open-source and can be found on GitHub at github.com/FiammettaC/ILP-user-preferences. ChatGMP is published and its code and data are available at github.com/FiammettaC/ChatGMP.

ACKNOWLEDGEMENTS

Funding was provided by (i) the Novo Nordisk Foundation, grant number NNF22OC0080136 - Real-time sustainability analysis for Industry 4.0 (Sustain4.0); (ii) Technical University of Denmark (DTU) and Novonesis (formerly Novozymes), Denmark; and, (iii) European Union under the Horizon Europe research and innovation programme with the grant agreement 101159993, within the HORIZON-WIDERA-2023-ACCESS-02-0 call.

REFERENCES

1. Caccavale, F., Gargalo, C. L., Kager, J., Larsen, S., Gernaey, K. V., & Krühne, U. (2024). ChatGMP: a case of AI chatbots in chemical engineering education towards the automation of repetitive tasks. *Computers and Education: Artificial Intelligence*, 100354.

2. Law, M., Russo, A., & Broda, K. (2019). Logic-Based Learning of Answer Set Programs. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11810 LNCS, 196–231. https://doi.org/10.1007/978-3-030-31423-1_6
3. Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. The Journal of Logic Programming, 19, 629–679.
4. Law, M., Russo, A., & Broda, K. (2015). Learning weak constraints in answer set programming. Theory and Practice of Logic Programming, 15(4-5), 511–525. https: //doi.org/10.1017/S1471068415000198
5. Law, M., Russo, A., Bertino, E., Broda, K., & Lobo, J. (2020). FastLAS: Scalable Inductive Logic Programming Incorporating Domain-Specific Optimisation Criteria. Proceedings of the AAAI Conference on Artificial Intelligence, 34(03), 2877–2885. https://doi.org/10.1609/aaai.v34i03.5678
6. Gelfond, M., & Lifschitz, V. (1988). The stable model semantics for logic programming. ICLP/SLP, 88, 1070–1080.
7. Erdem, E., Gelfond, M., & Leone, N. (2016). Applications of answer set programming. AI Magazine, 37(3), 53–68.
8. Agrafiotis, D. K., Bandyopadhyay, D., Wegner, J. K., & Van Vlijmen, H. (2007). Recent advances in chemoinformatics. Journal of Chemical Information and Modeling, 47(4), 1279–1293. https://doi.org/10.1021/ci700059g
9. Ando, H. Y., Dehaspe, L., Luyten, W., Van Craenenbroeck, E., Vandecasteele, H., & Van Meervelt, L. (2006). Discovering h-bonding rules in crystals with inductive logic programming. Molecular pharmaceutics, 3(6), 665–674.
10. Begam, B. F., & Kumar, J. S. (2012). A study on cheminformatics and its applications on modern drug discovery. Procedia engineering, 38, 1264–1275.
11. Muggleton, S. (1993). Inductive logic programming: Derivations, successes and shortcomings. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 667 LNAI(1), 21–37. https://doi.org/10.1007/3-540-56602-3_125
12. Lisi, F. A. (2007). Building rules on top of ontologies for the semantic web with inductive logic programming. arXiv preprint arXiv:0711.1814.
13. Costa, V. S., Fonseca, N. A., & Camacho, R. (2008). Logchem: Interactive discriminative mining of chemical structure. 2008 IEEE International Conference on Bioinformatics and Biomedicine, 421–426.
14. Kaalia, R., Srinivasan, A., Kumar, A., & Ghosh, I. (2016). Ilp-assisted de novo drug design. Machine Learning, 103, 309–341.
15. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S. and Webson, A., (2024). Scaling instruction-finetuned language models. Journal of Machine Learning Research, 25(70), pp.1-53.

© 2025 by the authors. Licensed to PSEcommunity.org and PSE Press. This is an open access article under the creative commons CC-BY-SA licensing terms. Credit must be given to creator and adaptations must be shared under the same terms. See https://creativecommons.org/licenses/by-sa/4.0/