A Proposal For Handling Query Ambiguity For Process Mining Tasks

Lucas Fortunato Das Neves[1], Chrysoula Zerva[2] and Alessandro Gianola[1]

[1]INESC-ID/Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
[2]Instituto de Telecomunicações/Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

Abstract

Process mining plays a critical role in extracting insights from organizational processes by analyzing event log data. However, it requires the engagement of domain experts and process analysts. While Large Language Models have enabled conversational agents for process mining tasks —reducing dependence on analysts — their adoption introduces new challenges. Users without an understanding of process mining often formulate ambiguous or ill-defined queries due to limited specific knowledge, hindering accurate analysis. To address this, we propose multiple approaches that can be further studied to reduce the ambiguity by relying on AI techniques like Retrieval-Augmented Generation and Chain-of-Thought. By reducing ambiguity, human interaction with conversational agents becomes more intuitive, further bridging the existing gap. We also introduce datasets Query-PM-LLM and AmbQuery-PM-LLM, which can be used as benchmarks for future conversational agents capable of solving process mining tasks.

Keywords

Business Process Management, Process Mining, Large Language Models, Conversational Agents, Ambiguity

1. Introduction

Business processes are structured "chains of events, activities and decisions" [1], where activities are performed by some actors, and decisions can involve these actors and possible object artifacts. They are the backbone of how organizations deliver value, whether it be producing goods, offering services, or achieving internal objectives. Business Process Management (BPM) [2] is a well-established research field whose main objective is to help organizations achieve their organizational goals. BPM provides techniques, formal methods, and tools to analyze, design, implement, monitor, and optimize business processes for greater efficiency and alignment with business objectives. To effectively map and communicate processes, organizations leverage process modeling languages, such as the BPMN standard.[1] Despite the advantages associated with modeling business processes, there exists a disconnect between how processes are documented and how they are actually executed, i.e., since the modeling may fail to capture inherent deviations of real-world workflows. Consequently, organizations also rely on data to uncover the actual process execution, leveraging automated techniques from process mining.

Process mining [3, 4] combines data science and process science to *automatically* extract from event logs information for the identification of inefficiencies, bottlenecks, and deviations. It can be both backward-looking, such as uncovering the root cause of a problem, and forward-looking, like predicting processing times or suggesting improvements [5]. One can exploit process mining tools and reasoners to infer process behaviors given the event log. Reasoners are tools that perform automated logical reasoning in order to solve process mining tasks. These include, but are not limited to, process discovery - based on the event log, discover a process - and conformance checking - comparing a given model and an event log to determine if there are any discrepancies between them: for both tasks various approaches

7th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis (OVERLAY 2025), October 26, 2025, Bologna, Italy
lucas.neves@tecnico.ulisboa.pt (L. F. D. Neves); chrysoula.zerva@tecnico.ulisboa.pt (C. Zerva); alessandro.gianola@tecnico.ulisboa.pt (A. Gianola)
0009-0004-1275-7913 (L. F. D. Neves); 0000-0002-4031-9492 (C. Zerva); 0000-0003-4216-5199 (A. Gianola)

[1]https://www.omg.org/bpmn/

exploiting symbolic reasoning have been proposed, such as using SAT [6] solving, Satisfiability Modulo Theories [7], and planning [8].

Large Language Models (LLMs) [9] have revolutionized AI, enabling advanced understanding and generation of human language, and facilitating interaction with humans in a seamless, more user-friendly manner. Hence, to allow domain experts to make better use of business processes without the assistance of process analysts and to bridge the gap among their different competencies, conversational agents that can solve specific process mining tasks have already been developed [10] [11].

The reliance on conversational agents leads to another set of challenges, since the inherent ambiguity of natural language can result not only in misinterpretation of user intents, but also in hallucinations or biased responses, further complicating the analysis [12]. For example, if a user prompts a model with "Could you enumerate the activities that occur in the Sepsis Cases - Event Log?", the model knows it has to provide a list of activities, however, if the user prompts the same model with "Tell me what happens in the sepsis data.", the model might be uncertain about the type of the expected answer (i.e., the user might desire the list of activities that happen in the Sepsis Cases - Event Log or a list of conformant traces that exemplify typical behavior). This parallel between the inherent flexibility and potential for multiple interpretations of natural language and formal systems has garnered interest even from philosophers, since they have always prioritized logical representation and precise meaning[2].

Aimed at disambiguating queries provided by non-expert users who may lack the technical knowledge, we propose a set of methodologies that can increase the robustness of conversational agents that assist in process mining tasks. This robustness will come from the deployment of AI techniques that can help disambiguate user queries, either by relying on additional information – *Retrieval-Augmented Generation Large Language Models(RAG-LLM)* – or relying on reasoning – *Using Chain-of-Thought(CoT).* In Section 3, we outline different pathways that generally start by identifying ambiguity once the user query is received, proceed to disambiguating the query if necessary, and finish by providing an answer. Given the multitude of ways LLMs are being applied in process mining and considering the multiple approaches we outline to disambiguate questions provided by users, the expected contributions are:

- **EC1:** A way to disambiguate the user query by relying on additional information through the use of *RAG-LLM.*
- **EC2:** Using *CoT* to reason from ambiguous query identification to desired answer.
- **EC3:** new datasets that can serve as a benchmark for future work involving tailored NLP tasks in the context of process mining.

2. Related Work

This Section presents a comprehensive overview of related work, beginning with contemporary issues surrounding ambiguities in conversational scenarios. Those are followed by relevant research regarding techniques like *CoT* and *RAG-LLM* capabilities. This Section ends by highlighting other work related to LLMs that can solve process mining tasks.

Ambiguity in LLM. Prior to the advent of LLMs, multiple works focused on addressing the challenge of ambiguity in natural language requirements for software development had already been developed [13] [14]. With the introduction of LLMs, the issue of ambiguity gained even more importance, considering that they have to deal with the ambiguities that arise in conversation scenarios. Aimed at solving ambiguities in conversations based on questions and answers, Abg-CoQA [15] focuses on identifying ambiguities (Ambiguity Detection), following up with clarification questions (Clarification Question Generation) - questions whose answers will eliminate ambiguity and allow a given model to answer the initial question that was previously ambiguous - and returning a final answer (Clarification-based Question Answering). While that paper presents an interesting approach to deal with ambiguity in conversational scenarios and therefore inspiring the present work to make use of additional questions to eliminate ambiguity, it does not focus on domain-specific contexts such as process mining. In our

[2] https://plato.stanford.edu/eNtRIeS/ambiguity/

methodology, we suggest a system component that, instead of asking clarification questions, can provide refined questions and let the user select the one that better represents their initial intent.

RAG & CoT. *RAG* allows LLMs to account for updated information and deal with hallucinations by integrating external knowledge sources, such as document repositories (e.g., Wikipedia) and search engines (e.g., Bing), outperforming state-of-the-art results in open-domain question answering as highlighted in [16]. However, challenges with indiscriminate context use have led to advancements like RQ-RAG [17], which refines queries for more effective retrieval through rewriting, decomposition, and ambiguity clarification. Instead of disambiguating a query that will later be fed to the retriever component, as is done in RQ-RAG [17], we suggest using the retrieved context itself to disambiguate the query, providing non-ambiguous options to the user. In parallel, *CoT* mimics human reasoning by breaking problems down into multiple sequential steps in order to solve them, mirroring the cognitive strategy of breaking down complex tasks into manageable, intermediate thoughts[3]. This is useful in the mathematical context. For instance, the application of *CoT* reasoning with bounded-depth autoregressive transformers to solve mathematical and dynamic programming problems is explored in [18]. The work of [19] explores enhancing the reasoning capabilities of LLMs using *CoT* prompting, a method that generates natural language rationales to guide problem-solving and leverages few-shot learning. Empirical evaluations of this study demonstrate *CoT*'s effectiveness in benchmarks like GSM8K [20], CSQA [21], and symbolic tasks; and gains as the model size increases. It can be hypothesized that *CoT* can effectively deal with ambiguity where the reasoning steps guide the transition from an ambiguous input to a refined and actionable response.

LLMs & Process Mining. Significant efforts have been dedicated to generating process models from textual descriptions, such as ProMoAI [22] and BPMN-ChatBot [23]. In the field of process mining, conversational agents like C-4PM [10] have been developed. When C-4PM [10] receives a user query, it identifies the user's intent and then transforms natural language input into a Linear Temporal Logic on Finite Traces (LTLf) formula [24] compatible with *Declare4py* using the help of GPT model. *Declare4py*, grounded on the Declare language [25], a declarative process modeling language, is a Python library that uses constraint-based specifications to analyze a process. There is also work [26] focused on assessing LLMs' understanding of process behavior and meaning of activities. [26] is oriented towards accomplishing three process mining tasks that leverage LLMs' general process knowledge and understanding of activity semantics: classifying process traces as anomalous or not, evaluating and classifying pairwise activity relations within traces, and predicting the next activity, incorporating semantic understanding. Both [10] and [26] still do not address the inherent ambiguity of natural language nor the ambiguity that can arise from limited knowledge within a given domain, which is extremely relevant if the goal is to give non-experts access to agents that solve process mining tasks.

3. Methodology

This section outlines the methodology we suggest for further research to address ambiguity in process mining queries. In Section 3.1, we specify the scope of ambiguity on which we focus and describe the creation of two datasets, **Query-PM-LLM** and **AmbQuery-PM-LLM**, based on the Sepsis Cases — Event Log, to support NLP tasks in process mining. In Section 3.2, we present a *RAG* component designed to retrieve relevant information and disambiguate queries. In Section 3.3, we propose a *CoT* component to facilitate query disambiguation through structured reasoning.

[3] https://www.ibm.com/think/topics/chain-of-thoughts

3.1. Datasets for NLP tasks in Process Mining

Considering the lack of tailored NLP tasks and benchmarking datasets in the field of process mining, as highlighted by [26], we prepared a dataset (**Query-PM-LLM**) based on the Sepsis Cases — Event Log[4]. This event log has over 1,000 cases with a total of 15,000 events, which required us to use a compact abstraction of it that is added to the dataset of the type <*input, answer*>, where the *input* comprises the event log abstraction and a non-ambiguous query. This abstraction, whose use was inspired by [27], corresponds to process flow (abstraction of directly-follows graph) and process variants (different sequences of activities within a process, each representing a set of traces that follow the same pattern), and both can be obtained using *pm4py*—a process mining library.

From the non-ambiguous queries present in that dataset, we used ChatGPT and Grok to generate ambiguous queries. Although multiple types of ambiguity can be considered, those that are most relevant in the context of conversational agents and specific domains, such as process mining, relate to:

- Ambiguities directly associated with natural language. In the work by [15], four types of ambiguity are highlighted, with two particularly relevant in this context: coreference resolution, which involves identifying what a pronoun refers to in a sentence; and answer types, which pertain to the uncertainty regarding the nature of the desired answer (e.g., when asked about a book, not sure if it is about identifying the title or genre of it).
- Ambiguities associated with domain knowledge, as highlighted in the context of Natural Language Requirements by [13], which aims to detect pragmatic ambiguity (ambiguity that arises from the reader's background knowledge).

We produced a second dataset (**AmbQuery-PM-LLM**) of the type <*input, answer*>, where the *input* comprises the abstraction and a generated ambiguous query, and *answer* corresponds to the answer of the respective non-ambiguous query. In Section 3.2 and Section 3.3, we motivate future work by outlining methods incorporating *RAG* and *CoT* to solve the ambiguities present in **AmbQuery-PM-LLM** and lead to non-ambiguous questions in this field, similar to those of **Query-PM-LLM**.

3.2. RAG Component

Building upon previous efforts mentioned in Section 2, it would be beneficial to enhance the user query handling process by first retrieving relevant external information. This could include event logs pertinent to the question or domain-specific information, such as industry standards or descriptions of concepts from the perspective of the domain, which helps reduce ambiguity. Following this retrieval, and as presented in Figure 1, the system could classify the user query as either ambiguous or non-ambiguous. For queries deemed non-ambiguous, the system could directly provide an answer. Conversely, for ambiguous queries, the system could leverage the retrieved information to map the query to potential non-ambiguous alternatives, prompting the user to select the one that is closely related to their initial intent (i.e., asking the user "Did you mean to ask Question X or Question Y?"). As the model receives the selected non-ambiguous query, it can directly provide an answer. Here, the space for research and exploration would arise by evaluating which kind of information retrieved could yield better disambiguations.

3.3. CoT Component

We further suggest experimenting with *CoT* in this field, considering that process mining tasks often involve multi-step logical reasoning, and query disambiguation can benefit from *CoT*. If the user query is already identified as ambiguous, which can be explored following the *RAG* methodology previously suggested, one can go from an ambiguous query to a non-ambiguous query and final answer.

Dataset. Dataset with reasoning steps that lead from the query to the answer (<*input, non-ambiguous query, answer*>, where the *input* corresponds to process abstraction and ambiguous query). Such a

[4]https://data.4tu.nl/articles/dataset/Sepsis_Cases_-_Event_Log/12707639

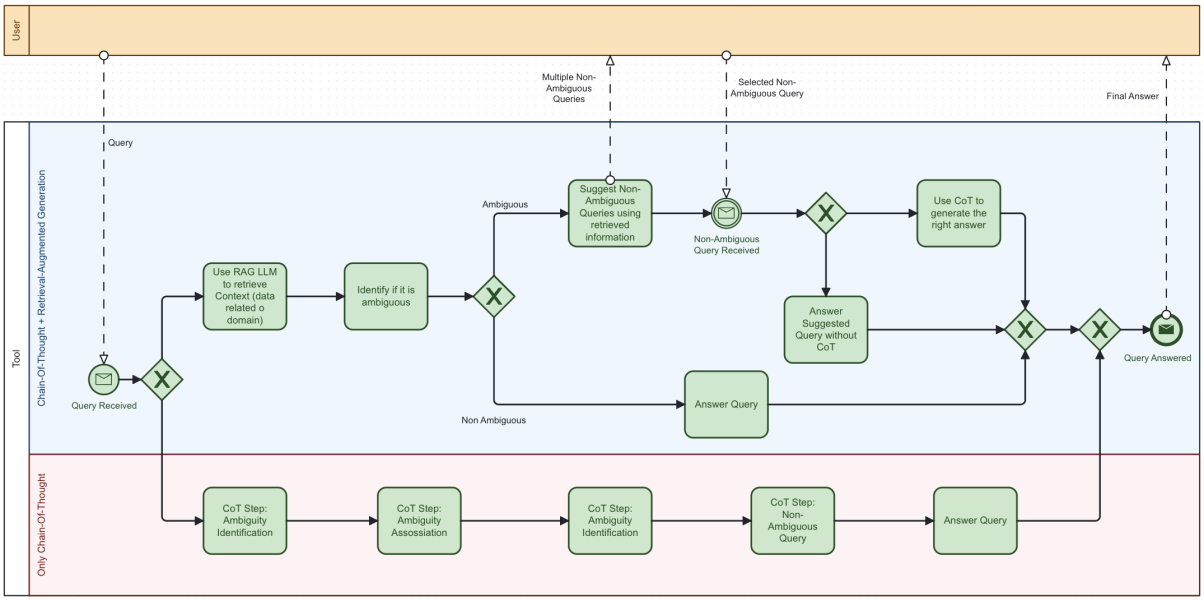

Figure 1: System Overview. The blue lane (up) considers *RAG + CoT* while the red lane (down) considers only *CoT*.

dataset can be created from **AmbQuery-PM-LLM**.

Another option which would not require the integration of other techniques like *RAG* as presented in Figure 1, corresponds to including ambiguity identification in the reasoning steps, leading to a dataset of the type: *<input, CoT Steps, answer>*, where *CoT Steps = <ambiguity identification, ambiguity association, non-ambiguous query>*, *ambiguity identification* confirms the existence of uncertainty and *ambiguity association* localizes the source of ambiguity within the question. The original datasets could also be used here in the same manner that was suggested for the other *CoT* approach.

3.4. Summary of Approaches and Their Integration

In summary, three research pathways can be explored:

- **RAG:** singularly explore the development of a *RAG* component as described in Section 3.2, using *RAG* for ambiguity detection, disambiguation, suggestion of non-ambiguous queries, and answer generation.
- **RAG + CoT:** use *RAG* for ambiguity detection, disambiguation, suggestion of non-ambiguous queries, and add *CoT* to use ambiguous and non-ambiguous queries for answer generation.
- **CoT:** use *CoT* to go from ambiguous query to answer generation, where the intermediate reasoning steps correspond to ambiguity identification, ambiguity association, and non-ambiguous query.

4. Evaluation

In this Section we present the results of one of the approaches previously outlined in Section 3.4. We implemented the approach corresponding to the exclusive use of *CoT* reasoning that goes from ambiguous query to answer generation.

4.1. Metrics

Semantic Entropy. To quantitatively assess the level of ambiguity in user queries and the effectiveness of using *CoT* for disambiguation, we employ semantic entropy as a key metric. Semantic entropy, introduced by [28], provides a measure of uncertainty in natural language generation that accounts for linguistic invariances, focusing on the diversity of meanings. Semantic entropy is computed as follows:

For a given query, multiple generations are sampled from a given *LLM* model. These generations are then clustered into semantic equivalence classes using a bidirectional entailment classifier (DeBERTa large model [29] fine-tuned with MNLI task[5]), which groups outputs that mutually entail each other (i.e., one can be inferred from the other, capturing shared meanings). The entropy is then estimated over the probability distribution of these clusters. Formally, if C represents the set of semantic clusters and Equation 1 is the semantic likelihood for cluster c given input x, then semantic entropy is given by Equation 2.

$$p(c \mid x) = \sum_{s \in c} p(s \mid x) \tag{1}$$

$$H_{\text{sem}}(x) = -\sum_{c \in \mathcal{C}} p(c \mid x) \log p(c \mid x) \tag{2}$$

In this specific experiment to compute semantic entropy, the model that we used for generation corresponds to Meta-Llama-3-8B-Instruct[6].

4.2. Results

The use of *CoT* drastically reduced the entropy on **AmbQuery-PM-LLM**, as can be seen in Table 1, indicating that the reasoning steps contributed to properly guide the model to generate answers based on the abstraction of the event log and disambiguate the user query when necessary.

Interestingly, the Average Semantic Entropy on **Query-PM-LLM**, while lower than the respective value for **AmbQuery-PM-LLM**, still is relatively high (0.996) suggesting that not only ambiguous queries, but also non-ambiguous ones can benefit from reasoning steps.

Table 1
Entropy Results

Dataset	Average Semantic Entropy	Number of Questions
Query-PM-LLM	0.996114	87
AmbQuery-PM-LLM	1.128019	148
AmbQuery-PM-LLM + *CoT*	0.413015	148

5. Conclusion

This paper tackles the significant challenge of query ambiguity in process mining tasks facilitated by *LLMs*. By suggesting the use of *RAG* and *CoT* techniques, we provide a pathway to resolve ambiguity and enhance the quality of interactions with conversational agents. Furthermore, we show that *CoT* positively contributes to the reduction of uncertainty.

The introduction of the **Query-PM-LLM** and **AmbQuery-PM-LLM** datasets further establishes a foundation for benchmarking and improving conversational agents. As the field evolves, future research can leverage our datasets and methodologies to create even more robust conversational agents, ultimately transforming how organizations understand and improve their processes.

Acknowledgments

This work was partially supported by the 'OptiGov' project, with ref. n. 2024.07385.IACDC (DOI: 10.54499/2024.07385.IACDC), fully funded by the 'Plano de Recuperação e Resiliência' (PRR) under the investment 'RE-C05-i08 - Ciência Mais Digital' (measure 'RE-C05-i08.m04'), framed within the financing

[5] https://huggingface.co/microsoft/deberta-large-mnli
[6] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

agreement signed between the 'Estrutura de Missão Recuperar Portugal' (EMRP) and Fundação para a Ciência e a Tecnologia, I.P. (FCT) as an intermediary beneficiary. This work was also partly supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT), under projects UID/50021/2025 and UID/PRR/50021/2025.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] M. Dumas, M. L. Rosa, J. Mendling, H. A. Reijers, Fundamentals of Business Process Management, Second Edition, Springer, 2018. URL: https://doi.org/10.1007/978-3-662-56509-4. doi:10.1007/978-3-662-56509-4.

[2] M. Weske, Business Process Management - Concepts, Languages, Architectures, 2nd Edition, Springer, 2012. URL: https://doi.org/10.1007/978-3-642-28616-2. doi:10.1007/978-3-642-28616-2.

[3] W. M. P. van der Aalst, Process Mining - Data Science in Action, Second Edition, Springer, 2016. URL: https://doi.org/10.1007/978-3-662-49851-4. doi:10.1007/978-3-662-49851-4.

[4] W. M. P. van der Aalst, J. Carmona (Eds.), Process Mining Handbook, volume 448 of *Lecture Notes in Business Information Processing*, Springer, 2022. URL: https://doi.org/10.1007/978-3-031-08848-3. doi:10.1007/978-3-031-08848-3.

[5] C. D. Francescomarino, C. Ghidini, Predictive process monitoring, in: W. M. P. van der Aalst, J. Carmona (Eds.), Process Mining Handbook, volume 448 of *Lecture Notes in Business Information Processing*, Springer, 2022, pp. 320–346. URL: https://doi.org/10.1007/978-3-031-08848-3_10. doi:10.1007/978-3-031-08848-3_10.

[6] M. Boltenhagen, T. Chatain, J. Carmona, Optimized SAT encoding of conformance checking artefacts, Computing 103 (2021) 29–50. URL: https://doi.org/10.1007/s00607-020-00831-8. doi:10.1007/S00607-020-00831-8.

[7] P. Felli, A. Gianola, M. Montali, A. Rivkin, S. Winkler, Data-aware conformance checking with SMT, Inf. Syst. 117 (2023) 102230. URL: https://doi.org/10.1016/j.is.2023.102230. doi:10.1016/J.IS.2023.102230.

[8] M. de Leoni, G. Lanciano, A. Marrella, Aligning partially-ordered process-execution traces and models using automated planning, in: Proceedings of the Twenty-Eighth International Conference on Automated Planning and Scheduling, ICAPS 2018, Delft, The Netherlands, June 24-29, 2018, AAAI Press, 2018, pp. 321–329. URL: https://aaai.org/ocs/index.php/ICAPS/ICAPS18/paper/view/17739.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, CoRR abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762. arXiv:1706.03762.

[10] Y. Fontenla-Seco, S. Winkler, A. Gianola, M. Montali, M. L. Penín, A. J. B. Diz, The Droid You're Looking For: C-4PM, a Conversational Agent for Declarative Process Mining., in: BPM (Demos / Resources Forum), 2023, pp. 112–116.

[11] U. Jessen, M. Sroka, D. Fahland, Chit-chat or deep talk: Prompt engineering for process mining, 2023. URL: https://arxiv.org/abs/2307.09909. arXiv:2307.09909.

[12] A. Keluskar, A. Bhattacharjee, H. Liu, Do llms understand ambiguity in text? a case study in open-world question answering, 2024. URL: https://arxiv.org/abs/2411.12395. arXiv:2411.12395.

[13] A. Ferrari, G. Lipari, S. Gnesi, G. O. Spagnolo, Pragmatic ambiguity detection in natural language

requirements, in: 2014 IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), 2014, pp. 1–8. doi:10.1109/AIRE.2014.6894849.

[14] A. Ferrari, A. Esuli, A NLP approach for cross-domain ambiguity detection in requirements engineering, Automated Software Engineering 26 (2019) 559–598. URL: https://link.springer.com/content/pdf/10.1007/s10515-019-00261-7.pdf.

[15] M. Guo, M. Zhang, S. Reddy, M. Alikhani, Abg-coQA: Clarifying ambiguity in conversational question answering, in: 3rd Conference on Automated Knowledge Base Construction, 2021. URL: https://openreview.net/forum?id=SlDZ1o8FsJU.

[16] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, CoRR abs/2005.11401 (2020). URL: https://arxiv.org/abs/2005.11401. arXiv:2005.11401.

[17] C.-M. Chan, C. Xu, R. Yuan, H. Luo, W. Xue, Y. Guo, J. Fu, Rq-rag: Learning to refine queries for retrieval augmented generation, 2024. URL: https://arxiv.org/abs/2404.00610. arXiv:2404.00610.

[18] G. Feng, B. Zhang, Y. Gu, H. Ye, D. He, L. Wang, Towards revealing the mystery behind chain of thought: A theoretical perspective, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 70757–70798. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/dfc310e81992d2e4cedc09ac47eff13e-Paper-Conference.pdf.

[19] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, volume 35, Curran Associates, Inc., 2022, pp. 24824–24837. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[20] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, J. Schulman, Training verifiers to solve math word problems, CoRR abs/2110.14168 (2021). URL: https://arxiv.org/abs/2110.14168. arXiv:2110.14168.

[21] A. Talmor, J. Herzig, N. Lourie, J. Berant, CommonsenseQA: A question answering challenge targeting commonsense knowledge, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4149–4158. URL: https://aclanthology.org/N19-1421/. doi:10.18653/v1/N19-1421.

[22] H. Kourani, A. Berti, D. Schuster, W. M. van der Aalst, Promoai: Process modeling with generative ai, in: K. Larson (Ed.), Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24, International Joint Conferences on Artificial Intelligence Organization, 2024, pp. 8708–8712. URL: https://doi.org/10.24963/ijcai.2024/1014. doi:10.24963/ijcai.2024/1014, demo Track.

[23] J. Kopke, A. Safan, Efficient LLM-Based conversational process modeling (2024). URL: https://drive.google.com/file/d/1EVl0SVKeyTnsw6pb59WgvL1K8Y88weRk/view.

[24] G. De Giacomo, M. Y. Vardi, Linear temporal logic and linear dynamic logic on finite traces (2013) 854–860.

[25] C. D. Ciccio, M. Montali, Declarative process specifications: Reasoning, discovery, monitoring, in: Process Mining Handbook, volume 448 of *Lecture Notes in Business Information Processing*, Springer, 2022, pp. 108–152. URL: https://doi.org/10.1007/978-3-031-08848-3_4. doi:10.1007/978-3-031-08848-3_4.

[26] A. Rebmann, F. D. Schmidt, G. Glavaš, H. van der Aa, Evaluating the ability of llms to solve semantics-aware process mining tasks, 2024. URL: https://arxiv.org/abs/2407.02310. arXiv:2407.02310.

[27] A. Berti, D. Schuster, W. M. P. van der Aalst, Abstractions, scenarios, and prompt definitions for process mining with llms: A case study, 2023. URL: https://arxiv.org/abs/2307.02194. arXiv:2307.02194.

[28] L. Kuhn, Y. Gal, S. Farquhar, Semantic uncertainty: Linguistic invariances for uncertainty estimation

in natural language generation, 2023. URL: https://arxiv.org/abs/2302.09664. arXiv:2302.09664.
[29] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL: https://arxiv.org/abs/2006.03654. arXiv:2006.03654.