Vectorizing the Company: Accessing multimodal data for Process Mining

Matthias Pohl[1], Yorck Zisgen[2][0000−0002−9646−2829], and Agnes Koschmider[2][0000−0001−8206−7636]

[1] ames wiring GmbH, Ursensollen, Germany
[2] University of Bayreuth, Bayreuth, Germany

Abstract. Process mining is a set of techniques that derive insights into business processes from recorded prior process executions stored in information systems. The introduction of Generative AI in general and Large Language Models in particular continues to impact this field. Accordingly, research on GenAI in process mining so far has mainly focused on extracting a business process model from natural language text descriptions. However, the benefit of accessing so far unused data in various non-text formats (such as images, PowerPoint, or Excel) that constitute a large portion of a company's knowledge pool has not been explored yet. Accessing this data could enhance and explain the results of process discovery, conformance checking, and process enhancement, and contribute to the success of process mining. In this paper, we propose VeCo, an open-source Python library that encapsulates and simplifies all tasks required to attach a vector database to a local LLM and vectorize multimodal data and multiple data formats so that LLMs can utilize them. We demonstrate how the added vectorized data from companies helps improve process mining results in three use cases.

Keywords: Generative AI · Large Language Model · Vector Database · Process Mining.

1 Introduction

- 1 Motivation:
  - Process Mining
  - Generative AI
  - Applications of Generative AI in Process Mining
  - Research Gap / Problem Definition
- 2 Use Cases:
  - Use Case 1 - **Process Discovery**: "What are my activities within the context of this process?"
  - Use Case 2 - **Process Enhancement**: Could extract process decision rules.
  - Use Case 3 - **Conformance Checking**: Eine Aktivität kommt einmalig vor und ist im Modell nicht enthalten. Ist das eine Abweichung? CC würde

ja sagen. Mit Dokument kommt heraus, dass es eine schriftliche Anweisung gegeben hat, die nicht ins Referenzmodell aufgenommen wurde. Das Ergebnis ist also, der Trace ist conforming und das Referenzmodell ist veraltet, also ein model skip, aber keine Abweichung im Sinne eines Fehlers.
  – Further applications: OCPD, OCPE, OCCC, OCPA (Performance Analysis), OCOS (Operational Support)
- 3 Goal of the paper:
  – '3.1 provide an importable open-source library (such as PM4PY) [footnote to repository] [3]
  – '3.2 PDF/Audio/XLS/database -> VeCo -> Vector Database -> Local LLM -> Insights
  – '3.3 short listing: import VeCo, connect to db, vectorize file
- 4 Structure of the paper

Process Mining is ... It has received widespread adoption in various domains, such as healthcare, production, and public administration. Since the release of GPT-3.5 by OpenAI [1], Generative AI (GenAI) and Large Language Models (LLMs) in particular have received a lot of attention. Most research on GenAI in Process Mining has focused on extracting business process models from natural language text, while also the utilization of LLMs as a natural language interface to Process Mining tasks and ...

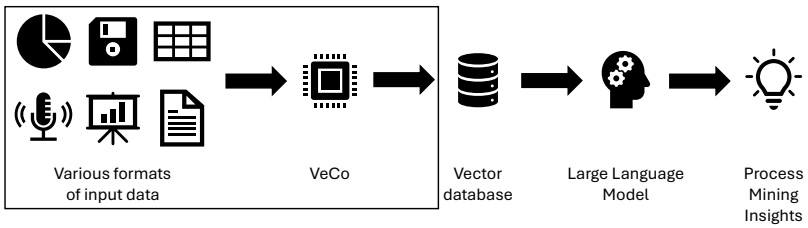

Fig. 1. zu 3.2 - Pipeline

[3] All code is available at https://github.com/MatthiasPohlAmberg/VeCo

2 Related Work

- works that deal with process modeling
- works that utilize domain knowledge
- works that use GenAI for natural language interaction
- research gap: nothing assists with vectorizing data to improve the application of LLMs to process mining tasks

Related to ours are works that *i)* extract business process insights from natural language textual descriptions, *ii)* that rely on present domain knowledge to improve the quality of results from Process Mining tasks, or *iii)* that utilize LLMs to accomplish natural language interaction with processes and data.

Extracting insights (predominantly business process models) from textual descriptions has been a mainstay of research since LLMs became widely accessible ([8], [9], [10]). ...

[6], [11], [12] ...

[1], [3], [7] ...

While the benefit of drawing on domain knowledge to improve the quality of process mining insights has been recognized, and the use of third-party libraries such as PM4PY [4] or PM4PY.LLM [5] has been widely accepted, tool support for the vectorization and LLM-accessible persistent storage of domain knowledge and its subsequent utilization for Process Mining purposes is still lacking.

3 Implementation

VeCo aims at providing a solution for the storage of a company's domain knowledge that is contained in many documents in a variety of file formats. From this, we conclude the following requirements for a solution:

- **Req 1:** A solution needs to support commonly used office formats, such as MS Word (.docx), MS Excel (.xls), MS PowerPoint (.ppt), Portable Document Files (.pdf), plain text files (.txt), as well as commonly used image types, such as .jpg, .jpeg, .png, and .bmp.
- **Req 2:** Domain knowledge can also be stored in animations, voice recordings, podcasts, or in video tutorials. Therefore, a solution would need to be able to handle file formats such as .wav or .mp3 for audio data, and .mp4 for video data.
- **Req 3:** Companies use a variety of proprietary and third-party applications that store information in specialized file formats not covered by the above. A solution should therefore be extendable to further file formats as and when the need arises.
- **Req 4:** A solution should be database- and LLM-agnostic, giving companies the choice of which models to use.

Fig. 2 shows the architectural composition of VeCo as a UML component diagram. It consists of components tasked with handling the user input (deciding which data format is given) and vectorizing the input accordingly. The Vectorize component relies on components to extract text from documents that contain text (such as .docx, .pdf, .txt, or .ppt), optionally compressing the content if desired (*Prompt: 'Please compress the following text: {raw_text}'*), embedding and indexing texts or textual descriptions of images and diagrams, and storing the vectorized input in a vector database. Further components provide the necessary transformation services,

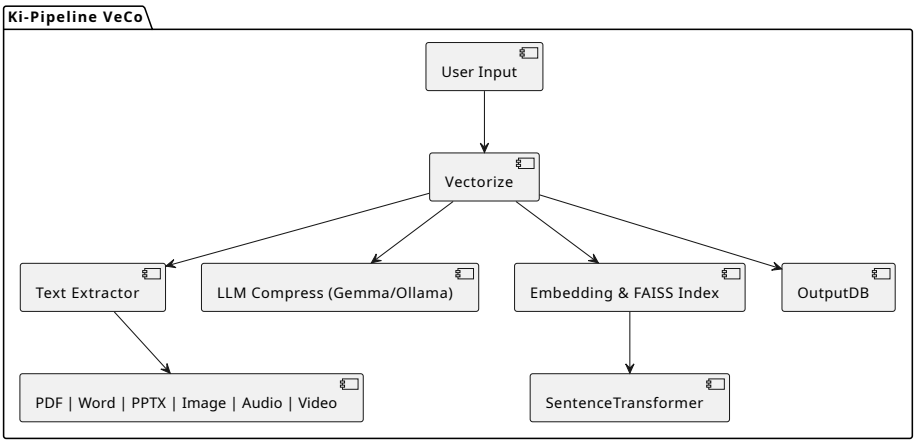

Fig. 2. UML Component Diagram

Fig. 3...

VeCo relies on several other Python packages, such as numpy and pandas. Notably, we use faiss for the search on the similarity of embedding vectors and OpenAI-whisper for the speech-to-text conversion. OpenAI-whisper relies on numba and llvmlite, which are currently incompatible with the Python 3.12 series. Therefore, our approach is reliant on the Python 3.10 series.

Further

...

4 Evaluation

- demonstrate for each use case:
  - show results without vectorizing
  - show listing with required code
  - show results with vectorizing
  - show why the second is better

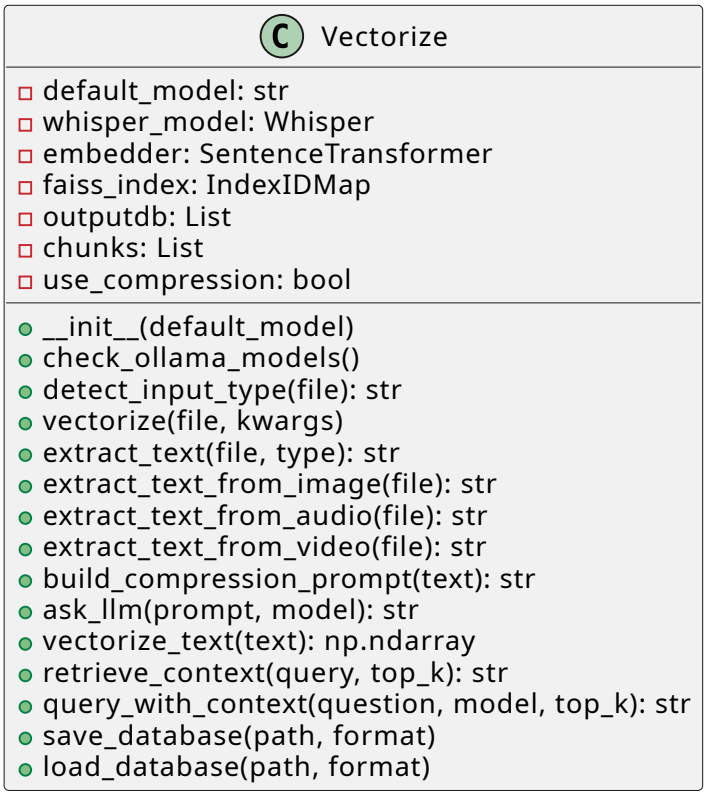

Fig. 3. UML Class Diagram

- reiterate how 'more domain knowledge' in the form of vectorized data provides 'better' process mining insights
- threats to validity
  - small sample size of just three examples -> purpose is to show the benefit of vectorizing domain knowledge
  - dependency of results may be influenced by choice of vector database and LLM -> out of scope of this paper

5 Conclusion

- GenAI in PM focuses on business process modelling
- while it is generally understood that more data results in better analysis results, no tool is yet available that supports including a company's domain knowledge
- we proposed VeCo, an open-source Python library
- we have presented two use cases and shown how vectorized domain knowledge supports and improves process mining techniques
- all code is available (cf. pg. 1)

References

1. W. M. P. Van Der Aalst, "Academic Perspective: How Object-Centric Process Mining Helps to Unleash Predictive and Generative AI," in Process Intelligence in Action, L. Reinkemeyer, Ed., Cham: Springer Nature Switzerland, 2024, pp. 219–232. doi: 10.1007/978-3-031-61343-2_22.
2. Apaydin, K., Zisgen, Y. (2025). Local Large Language Models for Business Process Modeling. In: Delgado, A., Slaats, T. (eds) Process Mining Workshops. ICPM 2024. Lecture Notes in Business Information Processing, vol 533. Springer, Cham. https://doi.org/10.1007/978-3-031-82225-4_44
3. Barbieri, L., Madeira, E., Stroeh, K., van der Aalst, W.M.P.: A natural language querying interface for process mining. Journal of Intelligent Information Systems pp. 1–30 (2022)
4. A. Berti, S. van Zelst, and D. Schuster, "PM4Py: A process mining library for Python," Software Impacts, vol. 17, p. 100556, Sep. 2023, doi: 10.1016/j.simpa.2023.100556.
5. A. Berti, "PM4Py.LLM: a Comprehensive Module for Implementing PM on LLMs," Apr. 09, 2024, arXiv: arXiv:2404.06035. doi: 10.48550/arXiv.2404.06035.
6. Dixit, P.M., Buijs, J.C.A.M., van der Aalst, W.M.P., Hompes, B., Buurman, H.: Enhancing process mining results using domain knowledge. In: Proceedings of the 5th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2015). pp. 79–94. CEUR-WS.org (2015)
7. Jessen, U., Sroka, M., Fahland, D.: Chit-chat or deep talk: prompt engineering for process mining. Working paper. arXiv.org (2023). https://doi.org/10.48550/arXiv.2307.09909
8. Klievtsova, N., Benzin, J., Kampik, T., Mangler, J., Rinderle-Ma, S.: Conversational process modelling: State of the art, applications, and implications in practice. In: Business Process Management Forum - BPM 2023 Forum, Proceedings. Lecture Notes in Business Information Processing, vol. 490, pp. 319–336. Springer (2023)

9. Kourani, H., Berti, A., Schuster, D., van der Aalst, W.M.P.: Process modeling with large language models. In: Enterprise, Business-Process and Information Systems Modeling - 25th International Conference, BPMDS 2024, and 29th International Conference, EMMSAD 2024, Proceedings. Lecture Notes in Business Information Processing, vol. 511, pp. 229–244. Springer (2024)
10. J. Neuberger, L. Ackermann, H. van der Aa, and S. Jablonski, "A Universal Prompting Strategy for Extracting Process Model Information from Natural Language Text Using Large Language Models," in Conceptual Modeling, W. Maass, H. Han, H. Yasar, and N. Multari, Eds., Cham: Springer Nature Switzerland, 2025, pp. 38–55. doi: 10.1007/978-3-031-75872-0_3.
11. A. Norouzifar, H. Kourani, M. Dees, and W. van der Aalst, "Bridging Domain Knowledge and Process Discovery Using Large Language Models," Aug. 2024, doi: 10.48550/arXiv.2408.17316.
12. Schuster, D., van Zelst, S.J., van der Aalst, W.M.P.: Utilizing domain knowledge in data-driven process discovery: A literature review. Comput. Ind. 137 (2022)