From Words to Workflows: Extracting Object-Centric Event Logs from Textual Data

Alina Buss[1][0009−0008−0371−8604], Christoph Kecht[2,3,4][0000−0002−1550−9841], Wolfgang Kratsch[2,3,5][0000−0001−9815−0653], Maximilian Röglinger[2,3,4][0000−0003−4743−4511], Sareh Sadeghianasl[6][0000−0002−0338−958X], and Moe T. Wynn[6][0000−0002−7205−8821]

[1] TUM School of Management, Technical University of Munich, Munich, Germany
✉ alina.buss@tum.de
[2] FIM Research Center for Information Management, Augsburg & Bayreuth, Germany
{christoph.kecht, wolfgang.kratsch, maximilian.roeglinger}@fim-rc.de
[3] Branch Business & Information Systems Engineering of the Fraunhofer FIT, Augsburg & Bayreuth, Germany
[4] University of Bayreuth, Bayreuth, Germany
[5] Technical University of Applied Sciences Augsburg, Augsburg, Germany
[6] Queensland University of Technology, Brisbane, Australia
{s.sadeghianasl, m.wynn}@qut.edu.au

Abstract. Organizations generate vast amounts of data in unstructured formats, such as textual descriptions, which remain largely untapped for process mining. This data is particularly valuable because it often captures critical exception cases and intricate dependencies that are absent in structured datasets, but crucial for understanding process deviations. Importantly, these unstructured sources frequently preserve the object-centric nature of real-world processes – information that is typically flattened or lost in traditional, case-centric event log formats. In this paper, we harness this potential and tackle the research gap by introducing a novel approach to extract Object-Centric Event Logs (OCELs) from unstructured textual descriptions using natural language processing techniques and large language models. Our approach consists of two subcomponents: a collector and a refiner. The collector aims to extract activities, timestamps, entities and their properties from textual descriptions, while the refiner integrates, cleans, and refines the extracted information from multiple descriptions. We implement both subcomponents in heuristic and generative forms, creating four distinct extractor variants that are compared against each other on synthetic textual descriptions derived from six publicly available OCEL datasets. Our results reveal that a generative collector combined with a heuristic refiner exhibits the strongest generalization capabilities on unseen textual descriptions.

Keywords: Object-Centric Event Logs · Process Mining · Natural Language Processing.

1 Introduction

Process mining aims to analyze and optimize business processes by deriving insights from real-world event data. The starting point of all process mining activities are event logs, which are detailed records of process events that capture the sequence and context of activities taken within a process. Given their foundational role, the accurate and comprehensive extraction of these event logs is paramount for the success of subsequent process mining procedures [12,13]. Most existing approaches focus on extracting event logs from structured data within organizations' core information systems [7]. However, an increasing amount of process-related data is generated outside these systems in unstructured formats. This data often emerges as a result of deviations from the expected process behavior, such as manual interventions or exception handling, and, therefore, captures valuable information absent in structured data sources. Consequently, the targeted application of Natural Language Processing (NLP) to extract event logs from unstructured data sources, as successfully demonstrated in extant research [4,7], enables a more comprehensive representation of real-world processes.

Furthermore, real-world processes often exhibit object-centric characteristics that are typically reflected in textual descriptions. For example, a recruitment process may involve multiple entities such as different applicants, applications, and vacancies [14]. Traditional case-centric event log formats like the eXtensible Event Stream (XES) standard [6] are unsuitable for representing relationships between these entities due to simplifying assumptions. To overcome this limitation, advanced Object-Centric Event Log (OCEL) formats such as the Object-Centric Event Data (OCED) meta-model [3] and the OCEL 2.0 format [2] have been proposed recently [14,15]. However, to the best of our knowledge, no existing approaches target the extraction of OCELs from unstructured textual data.

To address this research gap, we develop an approach that comprises two primary subcomponents: a **collector** that extracts activities, timestamps, entities and their properties from textual descriptions and a **refiner** that consolidates this information over multiple descriptions through data integration, cleaning, and refinement. Each subcomponent is implemented in both heuristic and generative forms, yielding four distinct combinations, referred to as extractor variants. We evaluate these variants on synthetic textual descriptions derived from six publicly available OCEL datasets, each containing 1,000 events. The results indicate that the most effective configuration combines a generative collector, which excels in semantic extraction, with a heuristic refiner that improves precision through clearly defined rules. On average, this hybrid extractor exhibits the strongest generalization capabilities on unseen data.

In summary, this paper makes two key contributions. First, it introduces a flexible approach for extracting OCELs from unstructured text. Second, it systematically compares four instantiations, highlighting the strengths and trade-offs of NLP techniques and Large Language Models (LLMs). The implementation of the extractor variants and the evaluation data are available on GitHub[1].

[1] `https://github.com/Alinabuss/OCEL-extractor`

2 Design and Development

Figure 1 illustrates our approach for extracting OCELs from unstructured textual descriptions. We aim to provide a generic and domain-independent solution by leveraging different NLP techniques, thus minimizing the need for human intervention and supporting automated extraction from large datasets. Our approach comprises two subcomponents: a **collector** and a **refiner**. Initially, the collector subcomponent iteratively processes textual descriptions of arbitrary length to extract relevant information and structure it into a preliminary OCEL format. Handling each description individually reduces overall execution time and supports the progressive addition of data. Next, the refiner subcomponent concatenates these preliminary snippets and aims to improve the overall coherence of the resulting OCEL by mitigating inconsistencies and redundancies arising from variations in data structures and terminologies that could lead to misinterpretations or an incomplete representation of the process.

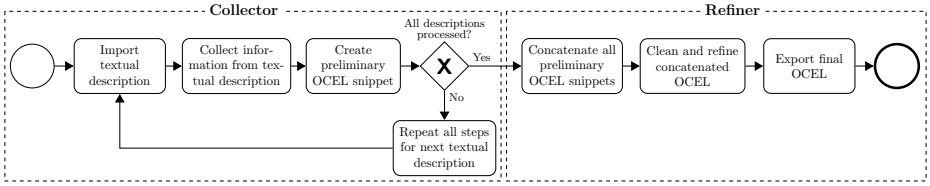

Fig. 1: Extraction approach

In the following, we instantiate the collector and refiner subcomponents in both heuristic and generative forms to extract event logs in the OCEL 2.0 format. The heuristic forms apply predefined rules, ensuring consistent outputs for the same input, while the generative forms utilize a LLM and thus allow for varying outputs for the same input. These subcomponents are combined into four distinct configurations, referred to as extractor variants: a HEU-HEU extractor (heuristic collector and refiner), a GEN-GEN extractor (generative collector and refiner), a GEN-HEU extractor (generative collector and heuristic refiner), and a HEU-GEN extractor (heuristic collector and generative refiner).

The **heuristic collector** gradually processes the provided textual descriptions using the Python NLP library SpaCy. A parsing pipeline tokenizes the text and extracts key token features, including dependency labels, Part-of-speech (PoS) tags, Named-entity recognition (NER) labels, and syntactic dependency relations such as children and ancestor tokens. Following a set of predefined rules, the collector evaluates the tokens, their dependencies, PoS tags, and NER labels to identify candidate values for the essential OCEL components: timestamps, activities, object labels, object types, attribute values, and attribute types. After refining the extracted values through lemmatization, analysis of their surroundings for reference values, and filtering redundant words extracted for multiple categories, these candidate values are assigned to OCEL components according

to predefined rules. Furthermore, by evaluating the associated children and ancestor tokens of each candidate value, as well as its positional context within the text, the collector maps object labels to object types, attribute values to attribute types, object labels to other object labels to reveal Object-to-Object (O2O) relationships, activities to timestamps, attributes to timestamps, object labels to activity-timestamp combinations to extract Event-to-Object (E2O) relationships, and attribute values to object labels and activity-timestamp-combinations. Based on these mappings, the heuristic collector generates a preliminary OCEL snippet per textual description. For example, the sentence "On January 15, 2023, the employee John Doe attended a training session" results in the following snippet:

```
{"objectTypes": [{"name": "Employee", "attributes": []}],
 "eventTypes": [{"name": "attend training session", "attributes": []}],
 "objects": [{"id": "John Doe", "type": "Employee"}],
 "events": [{
  "id": "1", "type": "attend training session",
  "time": "2023-01-15T00:00:00Z",
  "relationships": [{"objectId": "John Doe", "qualifier": null}]}]}
```

The **generative collector** invokes OpenAI's `gpt-4o-mini-2024-07-18` LLM and utilizes its included file-search capabilities. The textual descriptions are gradually provided via a user prompt to the LLM, which is then requested to generate a preliminary OCELs snippet per textual descriptions. To guide the extraction process and ensure adherence to the OCEL 2.0 format, an example event log containing a single event in this format is stored in the LLM's knowledge base. The corresponding system prompt can be found in the GitHub repository.

Afterward, within the **heuristic refiner**, the preliminary OCEL snippets are concatenated to a unified version that then undergoes a series of cleaning and refinement steps, leveraging predefined rules and majority-based approaches. These steps are repeated until the log attains a final state, with a maximum of five iterations. Within these iterations, the refiner alleviates data quality issues by, for example, resolving name inconsistencies, merging synonyms, and enforcing alignment between the `objectTypes`, `eventTypes`, `objects`, and `events` components of the OCEL 2.0 format. For the aforementioned example, a follow-up message could be: "January 18, 2023: John completed the training". From this text entry, several data quality issues emerge at the collector level. First, `"John"` will not be assigned to the `object` label `"John Doe"` and second, his `objectType` `"Employee"` will be missing since it wasn't restated explicitly. However, given the semantic similarity of `"John"` and `"John Doe"`, and the previously identified `objectType`, the heuristic refiner is able to resolve both issues.

In contrast, the **generative refiner** relies on an LLM, in our implementation again OpenAI's `gpt-4o-mini-2024-07-18` model. To this end, the concatenated event log is loaded into the LLM's knowledge base and the LLM is prompted to refine the event log and represent it in the OCEL 2.0 format. The corresponding user prompt can again be found in the GitHub repository.

3 Evaluation

We evaluate the four extractor variants using ground-truth data derived from six publicly available OCELs, allowing performance comparisons across multiple domains. Figure 2 depicts our evaluation framework, which comprises a **generator** instance, an **extractor** instance, and a **comparison** instance.

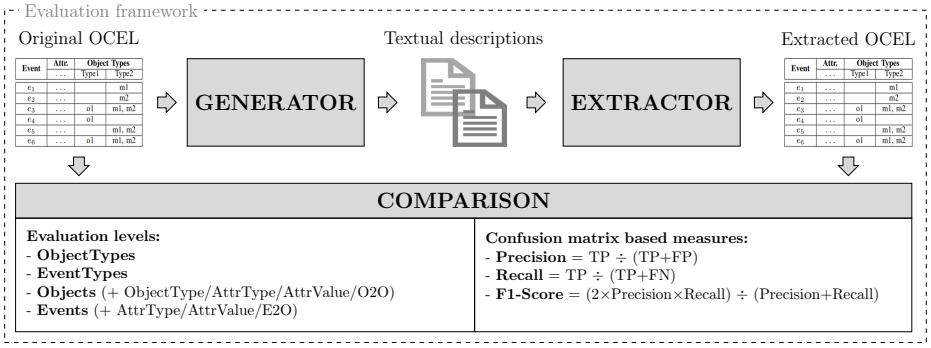

Fig. 2: Evaluation framework

Initially, we compile a dataset of six publicly available OCELs. Three of these logs – a recruitment log [1], logistics log [8], and Procure-to-Payment (P2P) log [11] – were previously employed in developing and validating the predefined rules in the heuristic collector and refiner. The remaining three logs – an order management log [9], a production log [5], and an Age of Empires log [10] – were not used during development, providing an opportunity to evaluate the generalization capabilities of each extractor variant. For each of the six event logs, we create a test subset of 1,000 events, ensuring there is no overlap between the test subsets and the subsets used during the development of the heuristic subcomponents.

The test subsets are then processed by the **generator instance**, tasked with converting the events into textual descriptions across three levels of complexity. One-third of the events is transformed into Complexity Level 1 descriptions – one textual description per event. Another third is converted into Complexity Level 2 descriptions – non-overlapping daily reports, with events grouped by day. The final third is transformed into Complexity Level 3 descriptions – overlapping reports, where events are grouped based on their related objects. To generate these textual descriptions, the generator instance utilizes OpenAI's `GPT-4o-mini-2024-07-18` model. The four extractor variants are then employed as the **extractor instance**, tasked with analyzing the provided textual descriptions to reconstruct the original OCEL. Each extractor variant leverages its respective heuristic or generative collector and refiner subcomponents to accomplish this task. As a result, one extracted OCEL in OCEL 2.0 format is created for each original OCEL.

Finally, the extracted OCELs are compared with their original counterparts using the **comparison instance**, which evaluates their alignment across various categories and levels of detail. The levels of detail – comprising parent and child levels – follow the structure of the OCEL 2.0 format. At the parent level, categories such as `objectTypes`, `eventTypes`, `objects`, and `events` are analyzed to ensure the existence of corresponding values in the extracted logs. Furthermore, at the child level, the comparison instance assesses whether specific child values are accurately mapped to their parent categories. For example, it verifies whether object types, attribute types, and attribute values are correctly linked to object labels, and whether the appropriate O2O and E2O relationships are identified. The overall score for each OCEL category is calculated by averaging the results across parent and child levels.

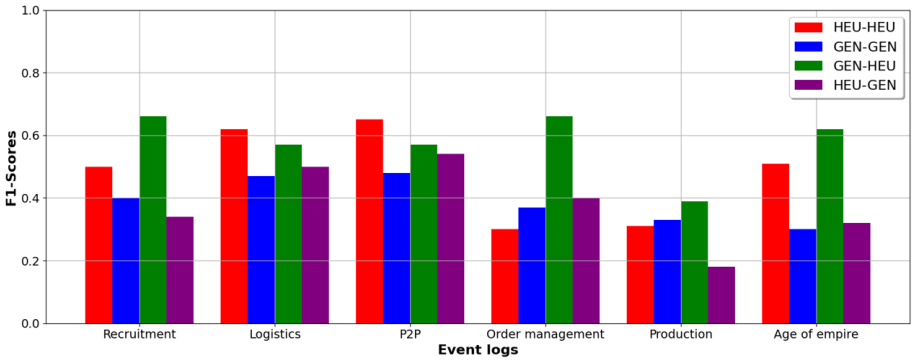

Fig. 3: Overall F1-score across all event logs

Figure 3 shows the overall F1-scores across all six event logs and all four extractor variants. The HEU-HEU extractor variant is particularly suitable for the three event logs used during the development of its heuristic subcomponents, surpassing all other variants on these three event logs, except for the GEN-HEU extractor variant on the Recruitment log. Although this finding suggests that the heuristic subcomponents were fine-tuned to the characteristics of the three event logs used during development, the HEU-HEU variant also shows promising generalization capabilities on the Age of Empires log, indicating that its performance is not strictly confined to the development data. In contrast, the GEN-GEN extractor variant achieves comparable, albeit more moderate results across all six event logs, which aligns with the fact that the LLM prompts were identical and not tailored to any specific log. However, the GEN-HEU extractor variant emerges as the overall best-performing approach, consistently outperforming both GEN-GEN and HEU-GEN variants on all six event logs, and exhibiting only minor performance differences between the development and test logs. Finally, the HEU-GEN extractor variant yields the lowest F1-scores on

average, particularly struggling on the test event logs and remaining consistently behind the GEN-HEU variant.

In conclusion, we recommend using the GEN-HEU extractor variant, which combines the strengths of the generative collector and the heuristic refiner. This hybrid approach consistently achieves the best overall performance, delivering satisfactory F1-scores and robust generalization capabilities. Furthermore, its completely unsupervised nature eliminates the need for human intervention, enabling automatic application across a wide variety of topics and large datasets. The generative collector component can also be fine-tuned with minimal effort by adjusting the LLM prompt, allowing the extractor to easily adapt to domain-specific requirements. However, it is important to acknowledge that our results are based on synthetically generated textual descriptions, which may not fully reflect data quality issues present in real-world texts, such as missing timestamps or inconsistent terminology. Additionally, our heuristic components showed reduced effectiveness when processing datasets significantly differing from those used during their development. Addressing these limitations through evaluation with real-world datasets should be the focus of future research to further enhance the reliability and accuracy of the extraction process.

4 Conclusion

This paper presents a novel approach for extracting OCELs from unstructured textual descriptions, thereby tackling a critical gap in process mining by incorporating textual data that often captures edge cases overlooked in structured data sources. Our approach comprises two distinct subcomponents – a collector and a refiner – that systematically transform textual descriptions into OCELs. Each subcomponent was instantiated in both heuristic and generative forms, resulting in four combined extractor variants that we compared against each other in an artificial evaluation on synthetic textual descriptions derived from six publicly available OCEL datasets. Our results reveal that a generative collector combined with a heuristic refiner exhibits the highest average F1-score and the strongest generalization capabilities on unseen textual descriptions.

The key contribution of our research is a flexible approach that systematically leverages NLP techniques and LLMs to enable process mining on unstructured text data. Specifically, our approach addresses critical gaps by handling object-centric data embedded in textual descriptions, which often include valuable insights on process deviations and manual exception handling. Furthermore, we systematically compare heuristic and generative methods. The implementation of the extractor variants and the evaluation data are available on GitHub. Future work should apply our approach to real-world textual descriptions to demonstrate its viability in practice. In parallel, establishing robust benchmarks that assess how effectively the extracted OCELs support process mining in practical scenarios remains a promising avenue for future research.

Acknowledgments. This work was supported in part by the Bavarian Research Foundation (Bayerische Forschungsstiftung) [grant number AZ-1550-20].

References

1. Berti, A.: Collection of object-centric event logs (OCEL 2.0 format; JSON specification) (2023). https://doi.org/10.5281/ZENODO.8433706
2. Berti, A., Koren, I., Adams, J.N., Park, G., Knopp, B., Graves, N., Rafiei, M., Liß, L., Tacke Genannt Unterberg, L., Zhang, Y., Schwanen, C., Pegoraro, M., van der Aalst, W.M.P.: OCEL (Object-Centric Event Log) 2.0 specification (2024), https://arxiv.org/abs/2403.01975
3. Fahland, D., Montali, M., Lebherz, J., van der Aalst, W.M.P., van Asseldonk, M., Blank, P., Bosmans, L., Brenscheidt, M., di Ciccio, C., Delgado, A., Calegari, D., Peeperkorn, J., Verbeek, E., Vugs, L., Wynn, M.T.: Towards a simple and extensible standard for object-centric event data (OCED) – Core model, design space, and lessons learned (2024), https://arxiv.org/abs/2410.14495
4. Geeganage, D.T.K., Wynn, M.T., ter Hofstede, A.H.: Text2EL: Exploiting unstructured text for event log enrichment. In: 2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Dijon, France. pp. 1–8 (2022). https://doi.org/10.1109/SITIS57111.2022.00010
5. Heinisch, M., Graves, N., van der Aalst, W.M.P.: sOCEL 2.0: A sustainability-enriched OCEL of a hinge production process (2024). https://doi.org/10.5281/ZENODO.13638681
6. IEEE: IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams (2016). https://doi.org/10.1109/IEEESTD.2016.7740858
7. Kecht, C., Egger, A., Kratsch, W., Röglinger, M.: Event log construction from customer service conversations using natural language inference. In: 2021 3rd International Conference on Process Mining (ICPM), Eindhoven, Netherlands. pp. 144–151 (2021). https://doi.org/10.1109/icpm53251.2021.9576869
8. Knopp, B., Graves, N.: Container logistics object-centric event log (2023). https://doi.org/10.5281/ZENODO.8428084
9. Knopp, B., van der Aalst, W.M.P.: Order management object-centric event log in OCEL 2.0 standard (2023). https://doi.org/10.5281/ZENODO.8428112
10. Liss, L., Elbert, N., Flath, C.M., van der Aalst, W.M.P.: Object-centric event log for age of empires game interactions (2024). https://doi.org/10.5281/ZENODO.13365584
11. Park, G., Tacke genannt Unterberg, L.: Procure-to-payment (P2P) object-centric event log in OCEL 2.0 standard (2023). https://doi.org/10.5281/ZENODO.8412920
12. van der Aalst, W.: Process mining: Overview and opportunities. ACM Transactions on Management Information Systems **3**(2) (2012). https://doi.org/10.1145/2229156.2229157
13. van der Aalst, W.: Process Mining: Data Science in Action. Springer, Berlin, Heidelberg, 2 edn. (2016). https://doi.org/10.1007/978-3-662-49851-4
14. van der Aalst, W.M.P.: Object-centric process mining: An introduction. In: Cerone, A. (ed.) Formal Methods for an Informal World: ICTAC 2021 Summer School, Virtual Event, Astana, Kazakhstan, September 1–7, 2021, Tutorial Lectures. pp. 73–105 (2023). https://doi.org/10.1007/978-3-031-43678-9_3
15. Wynn, M.T., Lebherz, J., van der Aalst, W.M.P., Accorsi, R., Di Ciccio, C., Jayarathna, L., Verbeek, H.M.W.: Rethinking the input for process mining: Insights from the XES survey and workshop. In: Munoz-Gama, J., Lu, X. (eds.) Process Mining Workshops. ICPM 2021, Eindhoven, Netherlands. pp. 3–16 (2022). https://doi.org/10.1007/978-3-030-98581-3_1