EasyChair Preprint

№ 9845

Discovering Process Models from Patient Notes

Rolf B. Bänziger, Artie Basukoski and Thierry Chaussalet

March 7, 2023

Discovering Process Models from Patient Notes

Rolf B. Bänziger[1[0000-0003-3356-8770]], Artie Basukoski[1[0000-0002-2670-8598]], and Thierry Chaussalet[1[0000-0001-5507-6158]]

[1] University of Westminster, London, UK
{r.banziger, a.basukoski, chausst}@westminster.ac.uk

Abstract. Process Mining typically requires event logs where each event is labelled with a process activity. That's not always the case, as many process-aware information systems store process-related information in the form of text notes. An example are patient information systems (PIS), which store much information in the form of free-text patient notes. Labelling text-based events with their activity is not trivial, because of the amount of data involved, but also because the activity represented by a text note can be ambiguous. Depending on the requirements of a process analyst, we might need to label events with more or fewer unique activities: two similar events could represent the same activity (e.g. screen referral) or two different activities (e.g. screen adult ADHD referral and screen depression referral). We can therefore view activities as ontologies with an arbitrary number of entries.

This paper proposes a method that produces an ontology for the activities of a process by analysing a text-based event log. We implemented an interactive tool that generates process models based on this ontology and the text-based event log. We demonstrate the proposed method's usefulness by discovering a mental health referral process model from real-world data.

Keywords: Process Mining, Text Mining, Healthcare, Mental Healthcare.

1 Introduction

Process Mining [1] is a set of techniques and tools to extract knowledge from event logs – traces left by process-aware IT systems describing who executed what activity and when. Event logs typically contain a list of events associated with a case. Each event is described by an activity (the process step it represents) and often includes additional data, such as date/time, the user involved and activity-specific information.

While the extracted knowledge can take many forms, Process Mining is often used to discover a process model as a flowchart that can be used to visualise and improve workflows. Many mature open and commercial Process Discovery algorithms emerged [2–6]. However, they all need event logs where each event is labelled with the activity. While this is usually the case in more structured systems and processes, such as a purchase-to-pay process managed in an ERP system, more flexible systems often do not include activity labels in their event logs.

Patient Information Systems (PIS) and their electronic patient records are an example of such flexible systems. Events are often documented using patient notes, i.e., free-text notes with no prescribed structure. Each event has to be labelled with its associated activity to discover a process model from such an event log. This is not a trivial task because the activity is not always obvious, especially without an apriori known list of activities. Allard et al. [7] present a multi-step manual workflow involving multiple to label a relatively small text-based event dataset. Clearly, the amount of work required to label an unlabeled event log becomes quickly prohibitive.

Furthermore, it is often not apparent whether two events represent the same activity or two different but related ones. Indeed, this may depend on the requirements of the process analyst using the process model. For instance, a process analyst analysing mental health data may be explicitly interested in patient journeys involving adult Attention Deficit Hyperactivity Disorder (ADHD). In this case, the process model needs to separate events related to adult ADHD from other events. However, if the analyst is interested in the high-level process flow, such a differentiation is not only unnecessary, it is detrimental to the comprehensibility of the resulting process models (process models with fewer activities are usually more accessible). Therefore, we must view activity labels not as a flat list but as a hierarchical ontology where the activity an event represents can be expressed using different (but related) concepts, depending on the required level of granularity of the process model.

We propose a method that discovers process models from text notes, leveraging an automatically created ontology describing activities. We implemented the method as an interactive tool and evaluated it with real-world data from a community mental health hub.

2 Background

While text data has been identified as an event log source, most of the research in Process Mining concentrates on using labelled event logs; far less research concerns itself with text-based event logs.

One of the first attempts at using free-text data as a basis for automated process discovery was described by [8]. They extracted email messages from Microsoft Outlook. Each email message had to be tagged manually by the Outlook user with the associated activity. They demonstrated the usefulness of applying Process Mining to text data but having users reliably tag emails (or other text notes) is often unrealistic or not feasible.

Jlailaty and Grigori [9] described a framework to extract business process activities from email logs using hierarchical clustering and compare the clustering quality using Latent Semantic Analysis (LSA) and Word2Vec. They did not suggest a method to label clusters automatically but instead relied on manual labelling.

We [10] proposed a framework to automatically extract process models from text notes stored in Customer Relationship Management (CRM) systems. We suggested using Latent Dirichlet Allocation (LDA) [11] to group notes representing the same activity and generate keywords for each group.

Chambers et al. [12] proposed a pipeline to discover business processes from emails. Their approach requires a ground truth dataset, as they use supervised machine learning and a ground truth excerpt of the data set to label emails.

de Medeiros et al. [13] gave an early overview of the opportunities of semantic Process Mining using ontologies. The same authors presented how using ontologies together with Process Mining can answer queries that regular Process Mining cannot. They provided a concrete implementation of their method.

To the best of our knowledge, there is no published approach that combines Process Mining from unlabelled text-based event logs with the semantic concepts of ontologies.

3 The Mining Process

Fig. 1 gives an overview of our method. We start by extracting an event log containing free-text patient notes. The text data and the positional information of each event are used to create two distance matrices. These matrices are combined and used as a basis for hierarchical clustering. We use the output of the hierarchical clustering, the dendrogram, as the ontology describing the process activities. We use feature selection techniques to label each node of the ontology. Finally, an interactive process miner shows the process model at various levels of detail selectable by the user, which can be exported for use in other Process Mining tools.

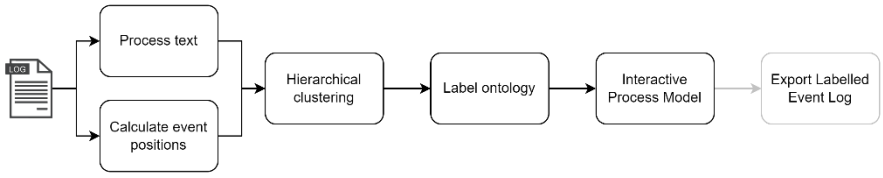

Fig. 1. Overview of the components of the mining process

3.1 Unlabelled event log requirements

In the first step, we extract an unlabelled event log from the source system (in our case, a patient information system). Each event has three fields: a case id, a timestamp and a free-text field containing the patient note. Timestamps allow us to determine the order in which events occurred. We are deliberately only looking at the minimum of information necessary to mine processes to prove the general viability of our approach. In practice, a process-aware information system might contain more information, e.g., the note's author, date, and other process-related information, which could be used to improve results.

3.2 Text processing

We are treating each event (or free-text note) as a document; all notes together make up our corpus. Each note is turned into a bag-of-n-grams [14]. We found that using n-

grams of length (1,2) provide a good compromise between quality of the resulting distance matrix and computing performance.

We create a document-term-matrix (*dtm*) containing term-frequency/inverse-document-frequency (*tf-idf*) values for each document/term combination. *tf-idf* values favour terms (or n-grams) occurring often in few documents and penalise terms occurring in many documents. This improves the eventual clustering by emphasising more important keywords while neglecting common words.

Finally, we use the cosine distance to calculate a distance matrix, yielding values between 0 and 1 for each pair of notes.

3.3 Calculating event positions

While text clustering works very well to identify certain activities, in other cases very similar notes can describe different activities. In our case, we have two activities that share similar vocabulary: a screening activity, where a clinician summarises the medical problem and an assessment, where another clinician writes about the medical situation of the patient in more detail. Naturally, both activities will use similar words. Using text data alone will put events from both activities in the same group. However, the activities are, in reality two different process steps that occur at different times.

As we are importing events in the order they occur, we know whether a note is at the start, the end, or anywhere in a process and can use this information to improve clustering results. We calculate two values for each event: distance from the beginning of the process and distance to the end of the process. Both values are scaled to fit between 0 and 1; then we use the Euclidean distance to create a distance matrix between each pair of events. Since we scaled values to (0,1), the distances use the same scales as the values in the matrix created from the text data.

3.4 Hierarchical clustering

In this step, we calculate the weighted average between both distance matrices. The weight is configurable in the interactive tool, however, giving both matrices the same weight seems to produce reasonable results. We expect that the more unique the vocabulary of each activity is, the less weighting should be given to the positional data.

We conduct hierarchical agglomerative clustering to generate a dendrogram. This dendrogram indicates which activities represent the same activity at different levels of detail. At the lowest level of detail, each event represents a unique activity, at the highest level of detail, each event represents the same activity. Since neither of these extreme levels of detail is useful, in practice, the user of the process mining tool will need to select a suitable level of detail.

3.5 Labelling the ontology

To turn the dendrogram of activities into an ontology, we need to label each node. We create a node-term-matrix, which contains the summed term-frequencies of all events/notes belonging to the respective node. We then calculate mutual information

for each node and term and select the six highest weighted terms of each node as the node label. Selecting six terms creates concise labels which allow users to infer the activity quickly.

3.6 Interactive Process Miner

Events following each other have a *directly follows* relationship. As our events are unique, each *directly follows* relationship is unique and therefore, all relationships have the same weight. When the user selects the level of detail of the process model, these relationships are aggregated through the activity ontology. The hierarchical ontology is "cut" at the specified level and the interactive process miner shows the process graph. It indicates frequencies by using bolder connecting arrows for common activities and transitions. It also provides interactive filters to hide infrequent activities and transitions.

This simple process discovery algorithm has some issues, the biggest being that it cannot detect parallelism. As there are many mature process discovery tools that might produce better result, our tool allows exporting the process as a regular, labelled event log.

4 Use Case

We are evaluating our process discovery tool using data from a London NHS trust providing mental health care. Part of this service are community adult mental health hubs. These hubs receive patient referrals from several sources, mostly general practitioners (GPs), but also social care, police, etc. Each referral must be assessed and referred to the appropriate secondary healthcare service.

This process is supposedly straightforward with only four activities: First, referrals are screened, and then admin staff books a tele-triage appointment with the patient. A clinician will assess the patient in the tele-triage appointment and finally refer and discharge the patient. After each of these activities, staff create a note in their PIS (see **Error! Reference source not found.** (a).

Referrals are supposed to be completed within two weeks. However, there are various problems with the process and sometimes referrals are abandoned due to patients not showing up, referrals not being appropriate, staff being overworked, etc. Since the process is mostly documented with patient notes, there is little visibility of the process. The hubs do not know how many processes deviate from the straightforward process, or when deviations occur.

We exported all patient notes that were created during a timespan of several months from one of the hubs. Each note is associated with an anonymised patient id. We also checked that notes do not contain patient names. After preparing the event log, we load it into our tool and choose appropriate parameters and filters. **Error! Reference source not found.** (b) shows one of the discovered process models, clearly showing the most frequent process pathways ("process highways") following the supposed process, but also showing infrequent deviations from the supposed process. We showed the process

model to subject matter experts, who were able to recognise their process, thus demonstrating that the technique is viable.

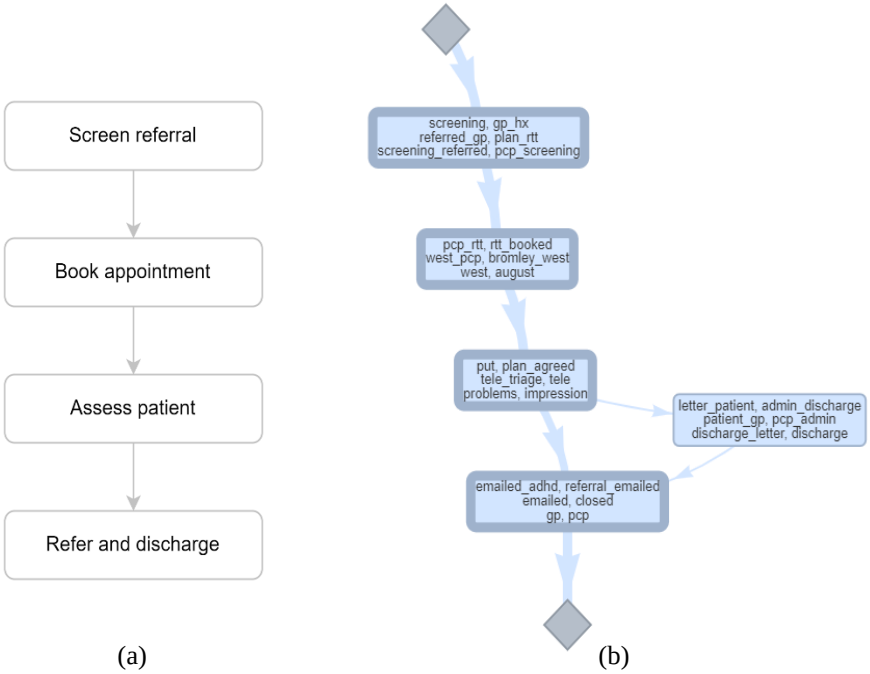

Fig. 2. (a) the supposed process (b) one of the discovered process models

5 Conclusion and future works

We introduced a novel approach to mine process models from text notes by using the concept of an ontology. Furthermore, we showed that using positional information can improve the clustering of events into activities. We demonstrated the viability of this approach using real-world clinical data.

In future research, we plan to use other available structured data, e.g., the note's author to further improve clustering. We also plan to evaluate text summarisation techniques to label the ontology.

References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19345-3.
2. Alves de Medeiros, A.K., Van Dongen, B.F., van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: Extending the α-algorithm to mine short loops. University of Technology. 113, 145–180 (2004). https://doi.org/10.1016/0076-6879(95)52025-2.

3. Weijters, A.J.M.M., van der Aalst, W.M.P., Medeiros, A.K.A.D.: Process Mining with the Heuristics Miner Algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP. 166, 1–34 (2006).
4. Günther, C.W., van der Aalst, W.M.P.: Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics. Business Process Management - Lecture Notes in Computer Science. 4714, 328–343 (2007). https://doi.org/10.1007/978-3-540-75183-0.
5. Leemans, S.J.J., Fahland, D., Van Der Aalst, W.M.P.: Process and deviation exploration with inductive visual miner. CEUR Workshop Proceedings. 1295, 46–50 (2014).
6. Engert, M., Chu, Y., Hein, A., Krcmar, H.: Managing the Interpretive Flexibility of Technology: A Case Study of Celonis and its Partner Ecosystem. (2021).
7. Allard, T., Alvino, P., Shing, L., Wollaber, A., Yuen, J.: A dataset to facilitate automated workflow analysis. PLOS ONE. 14, e0211486 (2019). https://doi.org/10.1371/journal.pone.0211486.
8. van der Aalst, W.M.P., Nikolov, A.: EMailAnalyzer: An E-Mail Mining Plug-in for the ProM Framework, (2007). https://doi.org/10.1.1.143.2975.
9. Jlailaty, D., Grigori, D.: Mining Business Process Activities from Email Logs. In: 2017 IEEE International Conference on Cognitive Computing (ICCC). IEEE (2017). https://doi.org/10.1109/IEEE.ICCC.2017.28.
10. Banziger, R.B., Basukoski, A., Chaussalet, T.J.: Discovering Business Processes in CRM Systems by leveraging unstructured text data. Presented at the The 4th IEEE International Conference on Data Science and Systems (DSS-2018) , Exeter, UK January 24 (2019). https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00257.
11. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research2. 3, 993–1022 (2003). https://doi.org/10.1162/jmlr.2003.3.4-5.993.
12. Chambers, A.J., Stringfellow, A.M., Luo, B.B., Underwood, S.J., Allard, T.G., Johnston, I.A., Brockman, S., Shing, L., Wollaber, A., VanDam, C.: Automated Business Process Discovery from Unstructured Natural-Language Documents. In: Lecture Notes in Business Information Processing. pp. 232–243. Springer Science and Business Media Deutschland GmbH (2020). https://doi.org/10.1007/978-3-030-66498-5_18.
13. de Medeiros, A.K.A., Pedrinaci, C., van der Aalst, W.M.P., Domingue, J., Song, M., Rozinat, A., Norton, B., Cabral, L.: An Outlook on Semantic Business Process Mining and Monitoring. In: Meersman, R., Tari, Z., and Herrero, P. (eds.) On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops. pp. 1244–1255. Springer, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76890-6_52.
14. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008).