INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

PROCESS MINING - RULE AUGMENTED ON SEQUENCES TO EVALUATE SUPERVISED EVENT ABSTRACTION RESULTS

Surendra Kumar Reddy Koduru, Research Scholar, Department of Computer Science and Applications, Monad University, Hapur, U.P.
Dr. Anil Badarla,Professor, Supervisor, Department of Computer Science and Applications, Monad University, Hapur, U.P.

Abstract: Process mining uses event logs to understand processes. Process discovery software often finds incoherent process models or models that do not match the log of events because events are not precise enough. We show that when process discovery methods find an inadequate process model from a lower-level event log, abstracting the log first may reveal the process's structure. This makes it difficult to create a structured process model from a low-level record of occurrences. We showed that supervised learning can help with event abstraction if high-level interpretations of low-level events are available for a subset of sequences (traces). Event extensions are used to generate features vectors. Conditional Random Fields with LSTM may abstract events from an event log utilising these event characteristics. We propose a sequence-based measure for supervised event abstraction evaluation that suits process identification and conformance-checking demands. The research concludes by showing how supervised abstraction of events may help organise and comprehend process models using real-life and synthetic event data.

Keywords—Process Mining, Rule Augmented, Event Abstraction, Conditional Random Fields

1 INTRODUCTION:

Process Mining software continually analyses and predicts execution flow to optimise business processes. It's cheaper and simpler. Process Mining analyses your business's back-end data objectively. In this competitive environment, businesses need frequent, straightforward input. To summarise, enterprises require an ideal analytic tool to

understand their development processes. Fig.1. Process mining is driven by showing how real-world processes work.

ISSN: 2057-5688

Software process mining automates transactional log process models. Mine these logs. Transactions represent cases or processes. Software business transactions are timestamps. Mining these timestamps creates a model visual flow. This approach helps us identify abnormalities in routine processes and understand their causes so we can optimise them.

Event logs are retrieved and prepared for method mining by wiping and reformatting the data. Way mining technologies offer more reliable technique flows than stakeholder interviews. This system-discovery situation provides data for future process evaluation and development, including way compliance and procedure path projection based on prior data.

Event logs are generally not created for process mining, hence their event granularity may be too low. Event logs must be abstracted for process discovery to work. Process discovery methods using low-level event logs may provide process models with undesirable features.

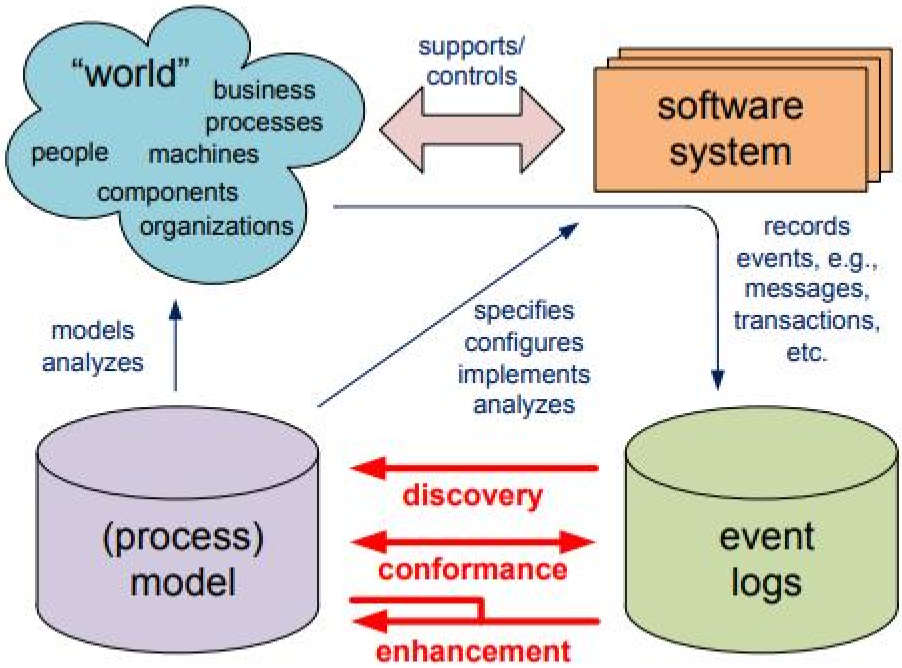

Figure 1 : Overview of Process Mining

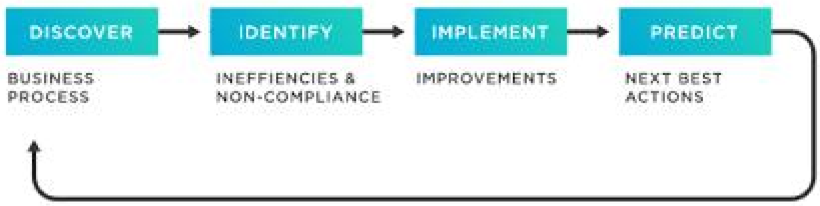

Figure 1 : Process Mining Enterprise System

This work provides supervised event abstraction for process detection from too finegrained event logs. Any event log with higher-level training labels of low-level events for a subset of traces may use this strategy. Section II reviews activity recognition-related work. Section III introduces key ideas and terminology utilised throughout the work. Section IV explains the challenge and proposes ways to automatically derive a feature vector representation of an event for supervised learning using the XES standard specification for event logs.

II RELATED WORK

2.1. Process Discovery Algorithms

Last decade saw many process discovery techniques. Alpha algorithms were the first process discovery techniques that automatically build Petri nets from event records. The original version [12] of the alpha algorithms may ensure the detection of particular behaviours in process models when the input event log is noise-free and meets certain completeness conditions. The original approach cannot find accurate process models with complicated behaviours. Later research studies refined the alpha approach to find short loops [13], invisible tasks [14,15], and non-free-choice behaviours [14,16]. Alpha algorithms work well on noise-free data, but they struggle when discovering process models from event logs. Alpha algorithms inspired the heuristics miners [17,18,19] to tackle event log noise.

Genetic algorithms [7,8], the ILP method [9], and machine-learning-based algorithms [3] are also used to develop process models. Most of these approaches find process models by sorting events in traces. Missing activity labels or events might mess up trace event sequencing [4].

The test benchmarked performance, managed inconsistencies, and examined variations of a bank customer service contact center's information handling incoming calls from consumers. The name centre event log revealed that the "over card restriction" issue had the highest number of inbound calls to the decision middle customer care phase. However, the results showed that nearly 32% of "Over Card Limit" issues were not solved on the first try (i.e., customers/customers had to re-dial and re-contact the operators in rate of the phase again for the second one), which can be used to improve customer support methods in a more green, effective, and timely manner.

This study uses process mining transition structures and regions to better understand a bank's Call Center's credit card customer care methods. We simulated and generated Transition System graphs using ProM process mining tool's best rules. This study's approach and recommended methods may inform future and unique investigations. Most decision centre customer support divisions of banks have systems to provide better customer support and statistics about public members of the agency's family to provide greater comfort to clients who use the issuer principal to boost performance in the banks' industrial organisation.

III. PRELIMINARIES

In this section we introduce basic concepts used throughout the paper.

As with any new technology, there are obvious mistakes that can be made when applying process mining in real-life settings. Therefore, we list six guiding principles to prevent users/analysts from making such mistakes.

Event Data Should Be Treated as First-Class Citizens

Starting point for any process mining activity are the events recorded. We refer to collections of events as event logs, however, this does not imply that events need to be stored in dedicated log files. Events may be stored in database tables, message logs, mail archives, transaction logs, and other data sources. More important than the storage format, is the quality of such event logs. The quality of a process mining result heavily depends on the input. Therefore, event logs should be treated as first-class citizens in the information systems supporting the processes to be analyzed. Unfortunately, event logs are often merely a “by-product” used for debugging or profiling.

Log Extraction Should Be Driven by Questions

Process mining activities need to be driven by questions. Without concrete questions it is

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

very difficult to extract meaningful event data. Consider, for example, the thousands of tables in the database of an ERP system like SAP. Without concrete questions it is impossible to select the tables relevant for data extraction.

IV EVENT ABSTRACTION AS A SEQUENCE LABELING TASK

4.1 Event abstraction

We assume that the techniques considered in this paper translate (multiple) instances of fine-granular events into instances of coarse-granular events, i.e., representing a level of detail that is closer, or equal to, the level of detail at which one aims to analyze the process. Our work is focused on event abstraction methods addressing the mapping from fine-granular events to coarse-granular events and, optionally, their connection to activity instances

Event abstraction as defined and considered in this paper can be seen in the broader spectrum of a larger hierarchy linking observations from the physical world to meaningful activity instances.

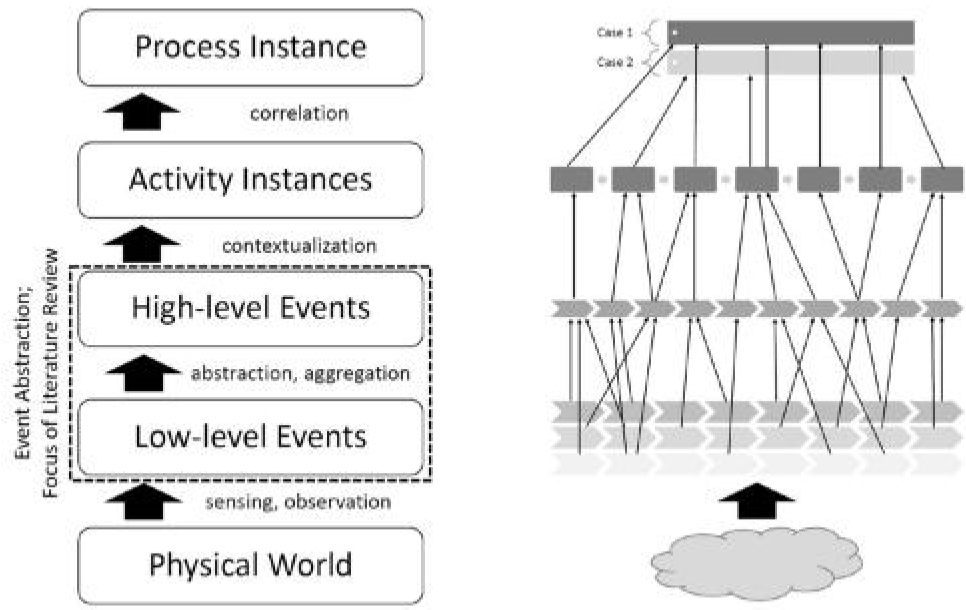

Figure 2: Hierarchy of mappings

4.2 Sequence labeling encoding

Given [x1, ..., xn] a sequence of n tokens, we encode event structures as token-level labels [y1, ..., yn], to reduce the task to a sequence labeling problem. Adopting dependency parsing terminology, we encode the label yi for each token xi as a tuple hd, r, hi, where d is the dependent and refers to the token and its

ISSN: 2057-5688

mention type (either trigger, entity, or nothing), r is the relation and used to refer to its role, and head (h) denotes the event the token refers to.

In more detail, to discriminate event heads with the same type in text, we encode the heads h as relative head mention position. For instance, h = +REG+1 means the head is the first +REGULATION on the right of d in the relative surface order, whereas h = +REG−2 means it is the second +REGULATION on the left

Word Embeddings

Due to the very nature of low resource languages, out-of-vocabulary words are common occurrences. To handle such words, we use pretrained fastText word embeddings and fine-tune on our corpus.

V EXPERIMENTS

5.1 Dataset

We release the new dataset named InDEE-2019 which consists of tagged event extraction data in disaster domain covering five languages: Marathi, English, Hindi, Bengali and Tamil.

For each language, we crawled disaster related documents from regional news websites. To annotate this documents we followed IOB (InsideOther-Beginning) tagging scheme. IOB tagging helps in differentiating between starting and ending of adjacent tags. In our disaster related documents, adjacent occurrence of same tags forms a phrase. Therefore, we adopted a simpler scheme that merges B and I together and tag under TO (T:Tag and O:Other) scheme. For each language, we divided dataset into three parts train (70%), validation (10%) and test (20%). The train dataset is used to train the model for event extraction task, validation is used for hyperparameter tuning and test dataset is used for testing our model.

Languages	Marathi(Mr)		Hindi(Hi)		English(En)		Tamil(Ta)		Bengali(Bn)	
	Doc	Sen	Doc	Sen	Doc	Sen	Doc	Sen	Doc	Sen
Train	825	16920	778	14178	475	5841	1478	17854	741	19854
Val	114	2024	140	2147	46	514	147	2417	154	7741
Test	244	4112	187	3641	147	1548	298	4127	187	3465
#Labels	41		39		45		41		39	

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

Table 1: Dataset for five languages, namely, Marathi, Hindi, English, Tamil and Bengali. Number of tags or labels for each dataset and their respective train, validation and test split used in the experiments.

VI RESULTS AND DISCUSSION

We use Bi-LSTM as our baseline and compare with the proposed three approaches. We conduct experiments on our In The dataset on 5 languages, namely, Hindi, Marathi, Bengali, Tamil and English. Our evaluation metric is standard micro-F1 and macro-F1 scores. Micro-F1 score counts the global true positives, false positives and false negatives whereas Macro-F1 captures the average unweighted class scores. Macro does not take class imbalance into consideration. We observe that due to highly skewed label distribution in our dataset, micro score is of more interest to us.

We train our model on varying training set size, namely, 20%, 40%, 60%, 80% and 100% to ascertain the impact of rules with decreasing amount of dataset. Most of the cases shows that our methods performs better than the baseline on less training data. Due to better word representation and data annotation on English, the parameters are even learnt on smaller dataset by Bi-LSTM. However, for low resource languages, Bi-LSTM is not able to learn parameters on similar training instances.

Tail Labels Most deep learning based methods are not able to capture tail labels due to lesser training data. However, this is of interest to us since most real world data has small size and has large label set. We only considered scores that are greater than baseline scores. We observe that more classes are improved by including rules over baselines. At lesser training instances, we observe that large number of tags are correctly classified. In order to further prove effectiveness of rule based approach, we tested the improvements of classes over tail labels. We chose those tails labels whose sum forms 5% of total training set instances. We notice significant improvement of tags over 20% and 40% training instances.

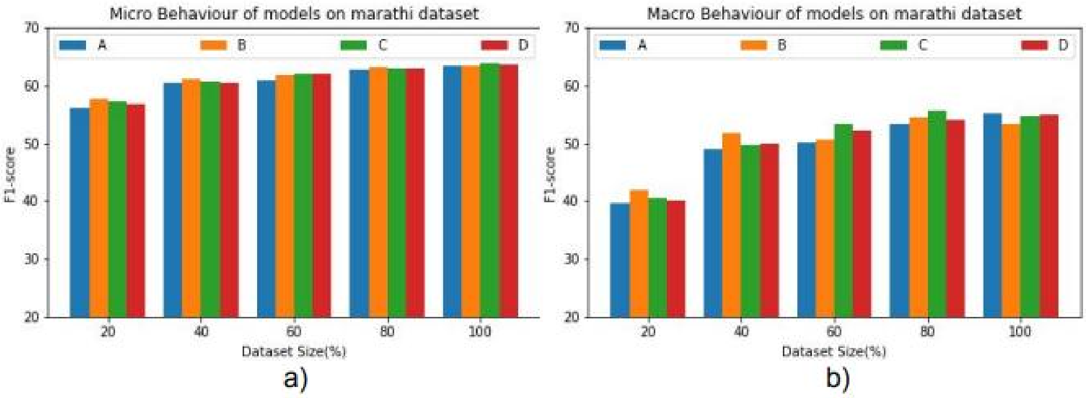

Figure : a) Comparison of Micro-F1 scores for different experiments over various training sizes (in %) b) Comparison of Macro-F1 scores for different experiments over various training sizes (in %)

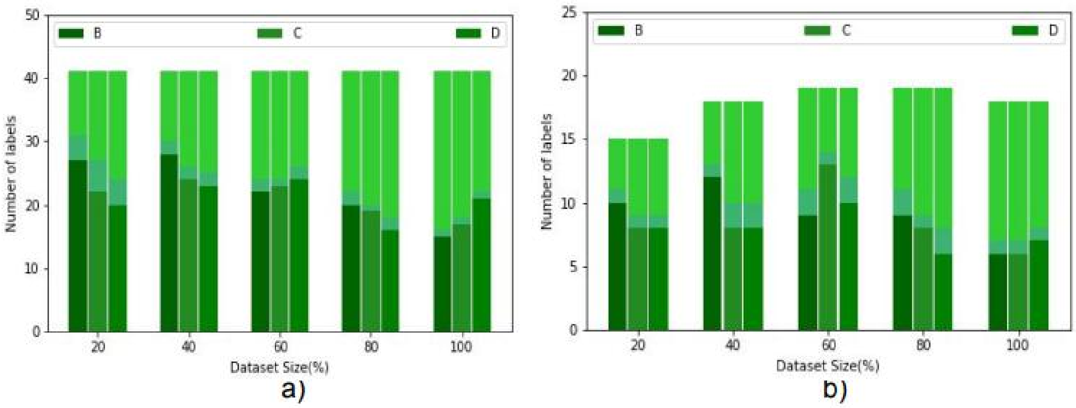

Figure : a) Comparison of proposed rule based approaches on improvement over all labels. b) Comparison of proposed approaches over tail labels. Lowest stack represents number of labels shown improvement over baseline, middle stack represent count of labels that has equal score with baseline and upper stack represent count of labels that has lesser score than baseline.

L	M	20	40	60	80	100
Mr	A	41.98	43.98	47.98	53.87	55.78
	B	45.76	**48.98**	49.81	54.8	51.80
	C	43.23	45.76	**55.98**	**57.23**	**58.90**
	D	42.12	44.38	49.67	51.87	52.23
Hi	A	31.23	43.49	49.12	48.32	47.99

	B	32.56	**47.45**	48.98	**49.76**	47.23
	C	35.09	42.89	50.23	48.23	**49.34**
	D	**30.12**	43.23	48.54	48.78	47.54
En	A	64.58	**73.7**	73.92	82.29	56.96
	B	61.65	63.66	**75.34**	75.31	82.71
	C	58.51	63.78	**74.47**	73.64	83.25
	D	63.85	62.24	71.67	**69.06**	79.01
Ta	A	44.88	**55.94**	59.64	62.15	62.62
	B	44.95	**57.55**	58.37	62.31	60.55
	C	44.36	55.75	**57.06**	64.08	62.73
	D	41.59	50.93	55.34	**60.75**	63.37
Bn	A	38.56	**49.69**	42.06	44.29	49.38
	B	42.92	**51.38**	38.26	47.57	48.3
	C	42.7	50.11	42.99	**47.65**	42.15
	D	41.41	46.54	39.34	42.33	**44.72**

Table 5: Comparison of Micro F1-score consisting tail labels for all 5 languages. L: Language, M: Models

VII CONCLUSION

In this paper, we introduce a hybrid approach for automatic extraction of events and arguments. We present a new dataset in the disaster domain for five languages consisting of large number of tags than usual datasets. We propose several variants of rule based system to augment deep learning based models. Extensive experimental results demonstrate that our rule augmented methods outperforms deep learning based models on lesser annotated data and low resource languages. We further shows more improvement on tail labels using our approach. For future work, we plan to integrate cross linking between events and its arguments.

VIII REFERENCES

[1] W. M. P. van der Aalst, Process mining: Discovery, conformance and enhancement of business processes. Springer Science & Business Media, 2011.

[2] W. M. P. van der Aalst, A. J. M. M. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1128–1142, 2004.

[3] C. W. Gunther and W. M. P. van der Aalst, "Fuzzy mining–adaptive ¨ process simplification based on multi-perspective metrics," in Business Process Management. Springer, 2007, pp. 328–343.

[4] J. M. E. M. Van der Werf, B. F. van Dongen, C. A. J. Hurkens, and A. Serebrenik, "Process discovery using integer linear programming," in Applications and Theory of Petri Nets. Springer, 2008, pp. 368–387.

[5] A. J. M. M. Weijters and J. T. S. Ribeiro, "Flexible heuristics miner (fhm)," in Proceedings of the 2011 IEEE Symposium on Computational Intelligence and Data Mining. IEEE, 2011, pp. 310–317.

[6] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering block-structured process models from event logs - a constructive approach," in Application and Theory of Petri Nets and Concurrency, ser. LNCS. Springer, 2013, pp. 311–329.

[7] R. P. J. C. Bose and W. M. P. van der Aalst, "Abstractions in process mining: A taxonomy of patterns," in Business Process Management, ser. LNCS. Springer, 2009, pp. 159–175.

[8] C. W. Gunther, A. Rozinat, and W. M. P. van der Aalst, "Activity ¨ mining by global trace segmentation," in Business Process Management Workshops, ser. LNBIP. Springer, 2010, pp. 128–139.

[9] B. F. van Dongen and A. Adriansyah, "Process mining: Fuzzy clustering and performance visualization," in Business Process Management Workshops, ser. LNBIP. Springer, 2010, pp. 158–169.

[10] C. W. Gunther and H. M. W. Verbeek, "XES-standard definition," ¨ BPMcenter.org, 2014.

[11] T. van Kasteren, A. Noulas, G. Englebienne, and B. Krose, "Accurate ¨ activity recognition in a home setting," in Proceedings of the 10th International Conference on Ubiquitous Computing. ACM, 2008, pp. 1–9.

[12] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in Pervasive Computing, ser. LNCS, A. Ferscha and F. Mattern, Eds. Springer, 2004, pp. 158– 175.

[13] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in Pervasive Computing, ser. LNCS, A. Ferscha and F. Mattern, Eds. Springer, 2004, pp. 1–17.

[14] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” ACM SIGKDD Explorations Newsletter, vol. 12, no. 2, pp. 74–82, 2011.

[15] R. Poppe, “A survey on vision-based human action recognition,” Image and Vision Computing, vol. 28, no. 6, pp. 976–990, 2010.

[16] L. Chen and C. Nugent, “Ontology-based activity recognition in intelligent pervasive environments,” International Journal of Web Information Systems, vol. 5, no. 4, pp. 410–430, 2009.

[17] D. Riboni and C. Bettini, “OWL 2 modeling and reasoning with complex human activities,” Pervasive and Mobile Computing, vol. 7, no. 3, pp. 379–395, 2011.

[18] T. van Kasteren and B. Krose, “Bayesian activity recognition in ¨ residence for elders,” in Proceedings of the 3rd IET International Conference on Intelligent Environments. IEEE, 2007, pp. 209–212.

[19] J. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: probabilistic models for segmenting and labeling sequence data,” in Proceedings of the 18th International Conference on Machine Learning. Morgan Kaufmann, 2001.

[20] L. R. Rabiner and B.-H. Juang, “An introduction to hidden Markov models,” ASSP Magazine, vol. 3, no. 1, pp. 4–16, 1986.

[21] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” Machine Learning, vol. 29, no. 2-3, pp. 131–163, 1997.

[22] E. Kim, S. Helal, and D. Cook, “Human activity recognition and pattern discovery,” Pervasive Computing, vol. 9, no. 1, pp. 48–53, 2010.

[23] W. Reisig, Petri nets: an introduction. Springer Science & Business Media, 2012, vol. 4.

[24] T. Murata, “Petri nets: Properties, analysis and applications,” Proceedings of the IEEE, vol. 77, no. 4, pp. 541–580, 1989.

[25] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. Van Dongen, and W. M. P. van der Aalst, “ProM 6: The process mining toolkit,” in Proceedings of the Business Process Management Demonstration Track, ser. CEURWS.org, 2010, pp. 34–39.