Analysis of Language Inspired Trace Representation for Anomaly Detection

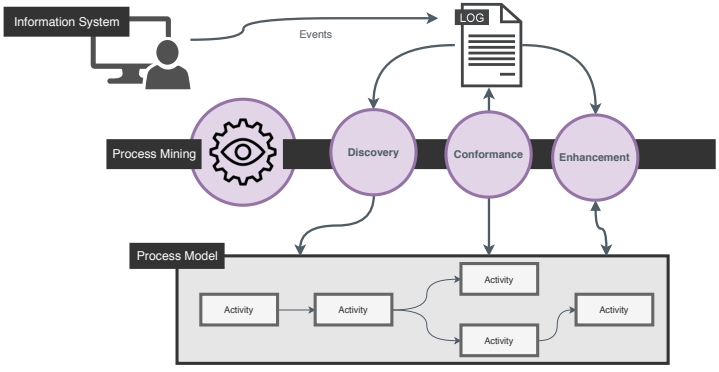
Gabriel Marques Tavares and Sylvio Barbon Jr.

Università degli Studi di Milano (UNIMI)
Londrina State University (UEL)

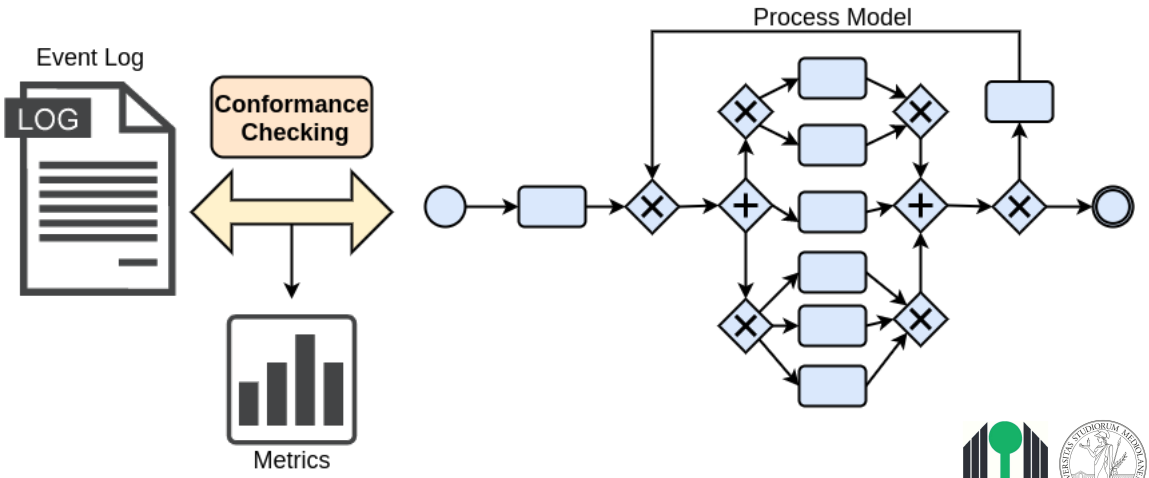
Introduction

Process Mining

"The idea of process mining is to discover, monitor and improve real processes by extracting knowledge from event logs readily available in today's systems" (Van Der Aalst, W. ,2011).



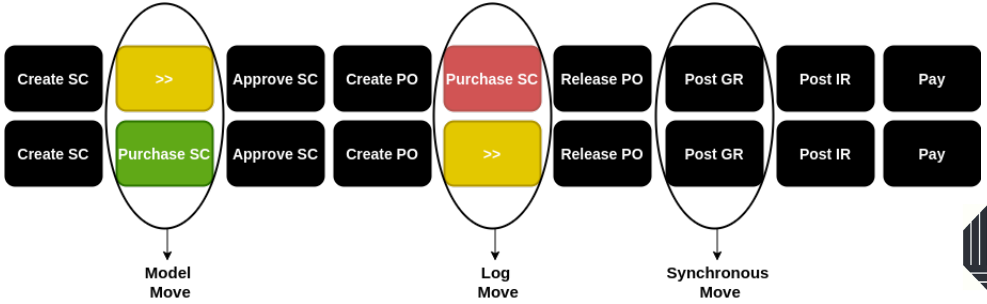
Conformance Checking



Classic Feature Engineering

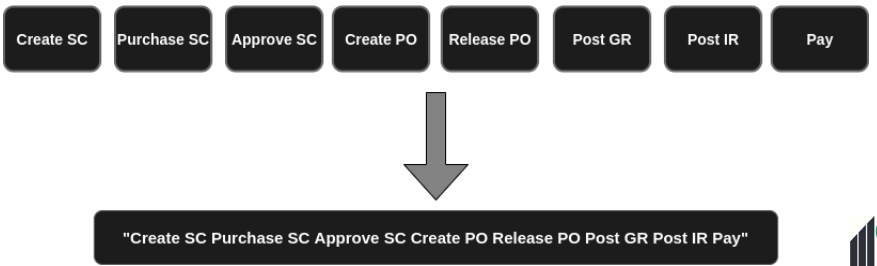
Token-replay: matches a trace to a process model and produces a fitness value along with counting tokens. Features: *trace_is_fit*, *trace_fitness*, *consumed_tokens*, *remaining_tokens*, *produced_tokens*.

Alignment: relates a trace to valid execution sequences in the model computing how synchronous they are. Features: *cost*, *visited_states*, *queued_states*, *traversed_arcs*, *fitness*.



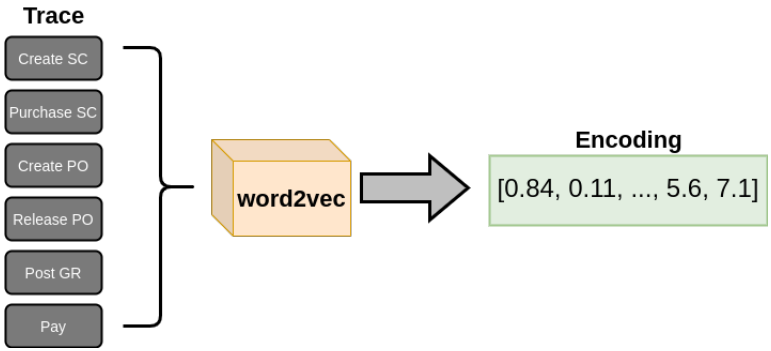
Word embeddings

- Process data contains several layers
- Encoding techniques can provide common grounds for analysis
- Activities describe the action being performed (i.e. words)
- Word embeddings capture context given a neighborhood



Word embeddings - word2vec

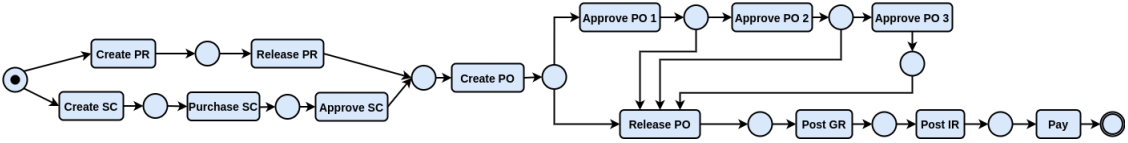
- Grounded natural language processing
- Weights of a two-layer neural network created to reconstruct the linguistic context of words in a corpus
- Words appearing in similar contexts generate more similar encodings
- Traces and activities are represented as sentences and words



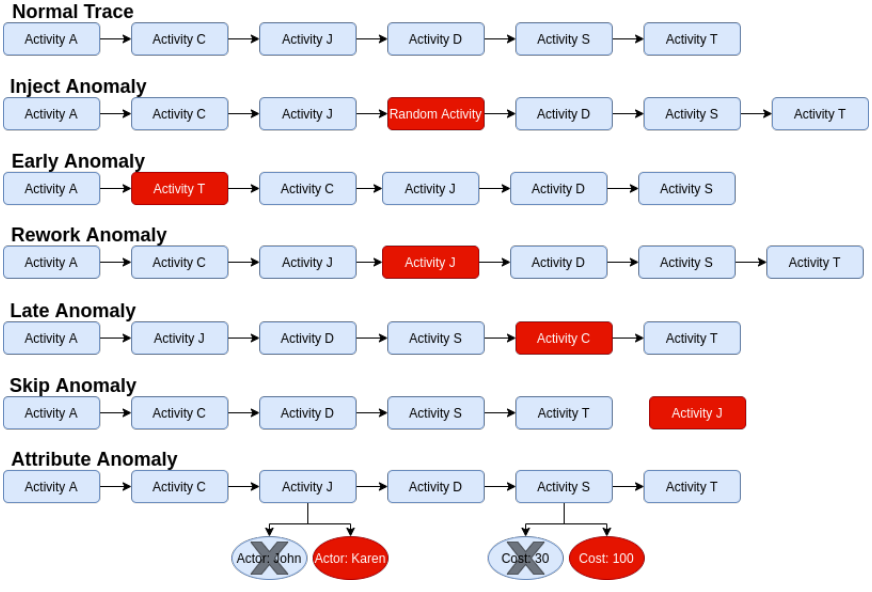
Experiments - Event Logs

Table: Event log statistics: each log contains different levels of complexity

Name	#Logs	#Activities	#Cases	#Events	#Attributes	#Attribute values
P2P	4	27	5k	48k-53k	1-4	13-386
Small	4	41	5k	53k-57k	1-4	13-360
Medium	4	65	5k	39k-42k	1-4	13-398
Large	4	85	5k	61k-68k	1-4	13-398
Huge	4	109	5k	47k-53k	1-4	13-420
Gigantic	4	154-157	5k	38k-42k	1-4	13-409
Wide	4	68-69	5k	39k-42k	1-4	13-382



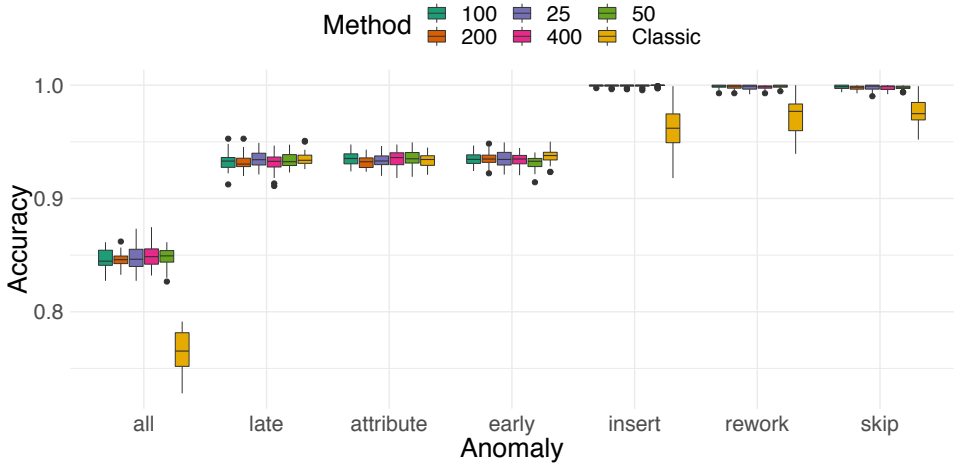
Experiments - Anomaly Types



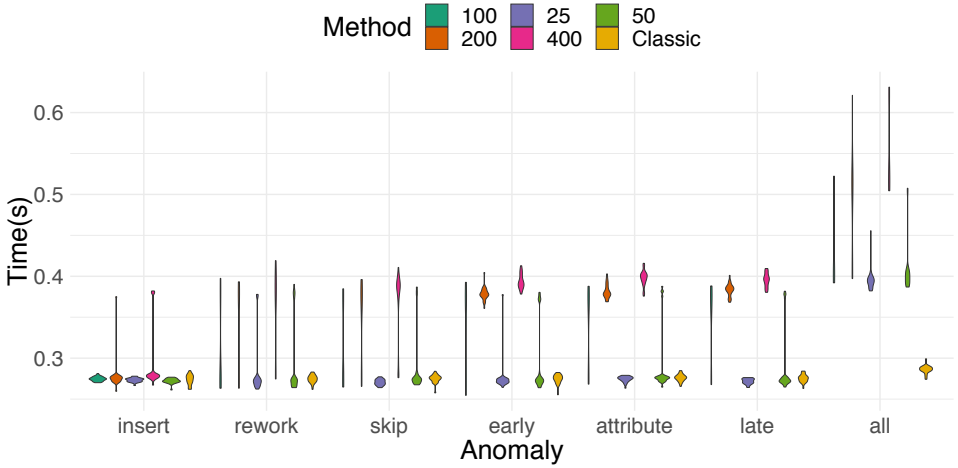
Experimental Setup

- Goals
  - Compare classic conformance checking and word2vec as encoding methods for business processes
  - Evaluate the impact of anomalies in the event log
- Classification
  - Binary: normal examples and one anomaly
  - Multi-class: normal examples and all anomalies
- word2vec encoding sizes: 25, 50, 100, 200, 400
- Random Forest (grid search)
  - *n_estimators*: 50
  - *max_features*: log2
  - *max_depth*: default
  - *entropy*: criterion

Results - Overall Performance

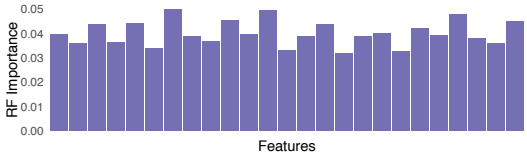


Results - Time Performance



Results - Feature Importance

Classic Features



Word2vec

Results - Anomaly Analysis

Results - Classic Features vs Word2vec

Table: Event log statistics: each log contains different levels of complexity

Task	Classic Features	Word2vec
All	76.3%	84.7%
Late, attribute, early	93.5%	93.3%
Insert, rework, skip	96.9%	99.8%
Time (all)*	0.28s	0.39s
Feature size	10	25, 50, 100, 200, 400
Feature importance	Alignment (4 features)	Distributed
Interpretability	✓	✗

* Classification time only

Conclusion

- Trace encoding based on word embeddings
- Word2vec performs better than traditional conformance features in several scenarios
- Anomalies impact encoding quality
- Future Work:
  - Consideration of multiple perspectives (e.g. time and resource)
  - Online encodings
  - Measuring encoding quality

Conclusion

Thank you!

github.com/gbrltv/business_process_encoding

gabriel.tavares@unimi.it