Discovering causal links in log files using machine learning

Vithor Bertalan

Polytechnique Montréal

DORSAL Laboratory

Introduction

- Machine learning is very good with labeling and clustering data
- But not as good with tracking the root cause of data
- Therefore, we can we do to identify data responsible for creating other data?

Object of Study

- Log files contain fundamental information about the execution of systems
- Modern software systems routinely generate a large volume of logs
- How to know which of the logs have a higher potential of causing further problems?

Causal Structure Learning

- A new field emerges: Causal Structure Learning
- Causality inference is basically *identifying the effect of an experimental intervention given observational data or a combination of observational and interventional data*. (Kalisch, 2014).
- The main goal of the area is to track the consequence of inserting/altering a single piece of data in a dataset.

First Hypothesis

- After log parsing, the *variable tokens* carry the most important information to track causality among log lines.

First Hypothesis

- Thus, we have to develop/adapt a technique to save variable tokens, instead of replacing them
- However, the most popular log parsing techniques use small inputs, instead of processing the whole text as a batch
- Using Transformers might be a good alternative

Parsing Method

1. Pre-processing raw log texts, using state-of-the-art tokenization techniques (e.g., SentencePiece)

2. Creating transformer embeddings from raw log texts

3. Clustering data into smaller subsets

Why cluster the logs?

- Separating common log sequences

Frequency Count

- After clustering, we count the frequencies of tokens in each cluster.
- Then, we define a threshold that separates static fields from variable tokens.

$$rf(i) = \frac{f(i)}{max_{i \in C} f(i)},$$

Error null pointer exception pkg2

Error null pointer exception pkg3

Token	Cluster	Frequency	Relative Frequency
Error	2	2	1
Null	2	2	1
Pointer	2	2	1
Exception	2	2	1
Pkg2	2	1	0.5
Pkg3	2	1	0.5

Second Hypothesis

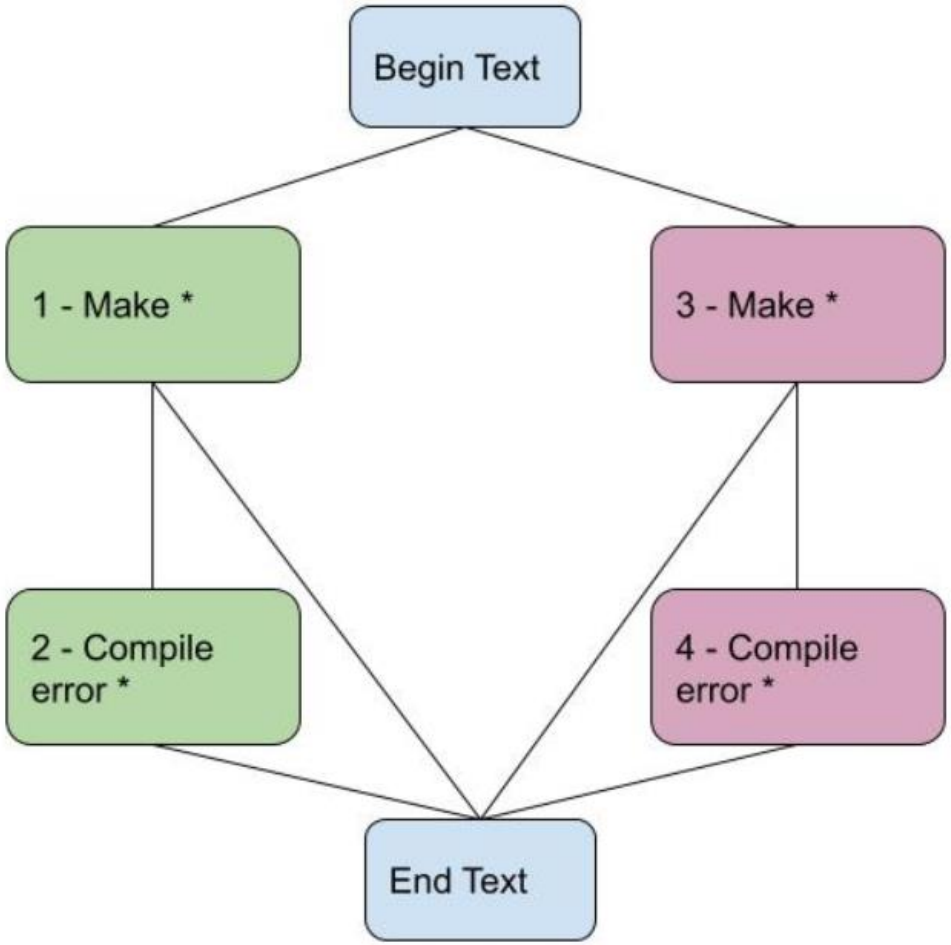
- In a graph, we presume that the most central nodes are the most relevant to the occurrence of descendant ones.
- Thus, we have to generate a graph from our log lines and track the centrality of each of the nodes.
- However, creating graphs from non-natural languages is not trivial.

Second Hypothesis

- We can use the parsed logs from the previous step to create our graphs. First, we parse:

Second Hypothesis

- Then, we connect the elements with common variable tokens, and track centrality.



Third Hypothesis

- Adopting CSL, using the parsed logs and the centrality of log lines created from previous steps as possible inputs.
- CSL shows good results in graphs, detecting arcs (directed vertices) between nodes.
- Our final goal: a matrix of probabilities of causalities between each pair of nodes.

References and Questions

- Kalisch, Markus, and Peter Bühlmann. "Causal structure learning and inference: a selective review." Quality Technology & Quantitative Management 11.1 (2014): 3-21.