Identifying Features of Android Apps from Execution Traces

Qi Xin, Farnaz Behrang, Mattia Fazzini, and Alessandro Orso

Understanding a Program & its Features

Refactoring

Debloating

Debugging

Functionality Modification

Testing

Documentation

Understanding a Program & its Features

Identifying Features of a program by Analyzing its Executions

Program Understanding is HARD

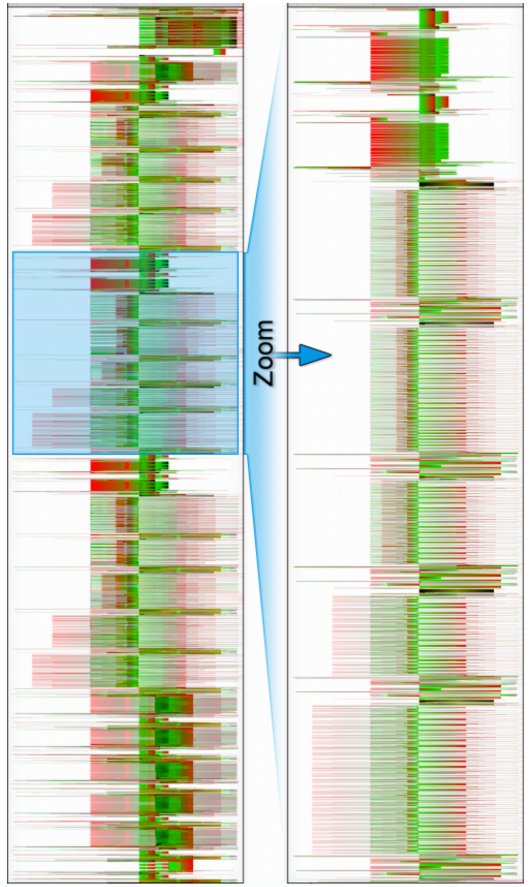


Figure from Understanding Execution Traces Using Massive Sequence and Circular Bundle Views by Cornelissen et al.

Program Understanding is HARD

*Program understanding: Challenge for the 1990s by Corbi

Figure from Understanding Execution Traces Using Massive Sequence and Circular Bundle Views by Cornelissen et al.

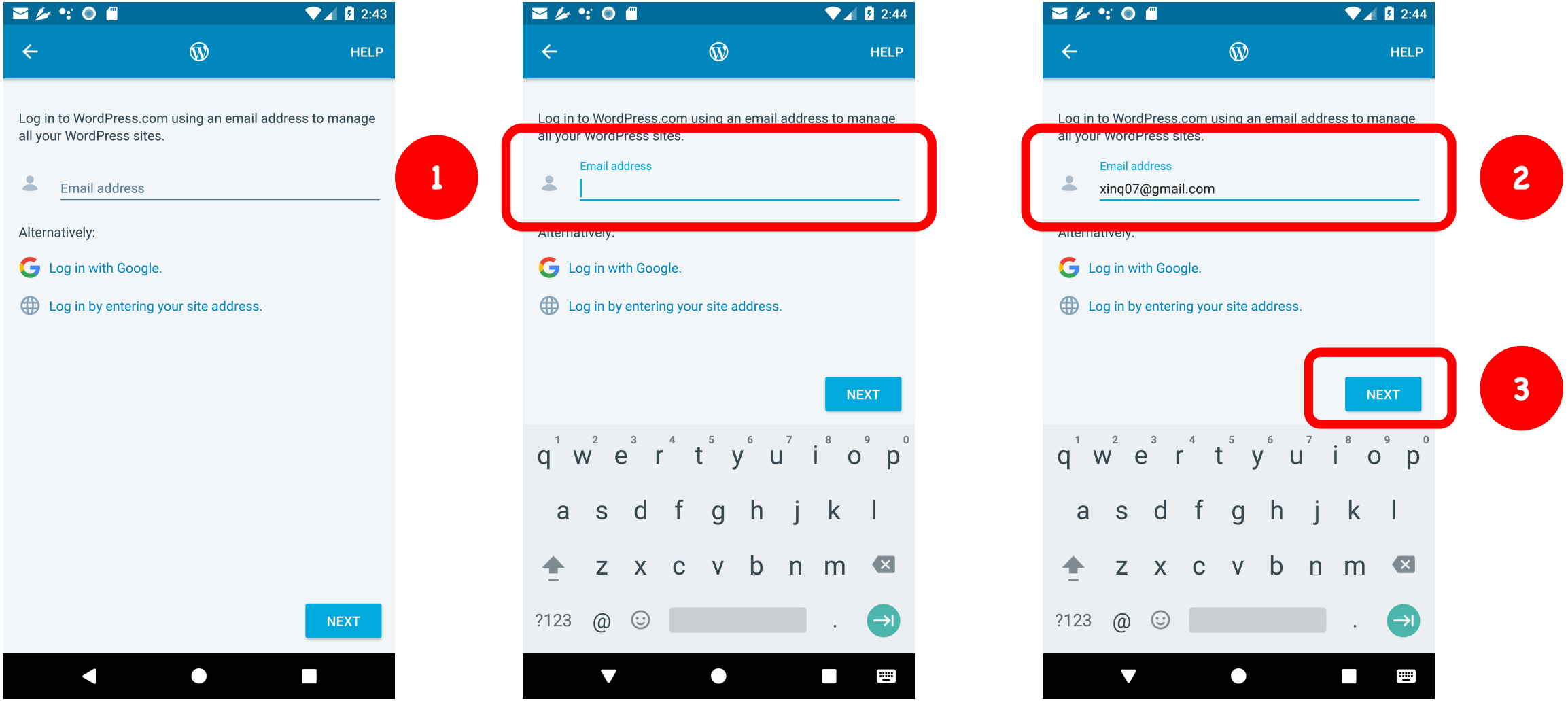
Our Approach

- Identifies features by analyzing execution trace
- Targets Mobile (Android) apps
- In our context, a feature is a sequence of user events that exercise some functionality of the app

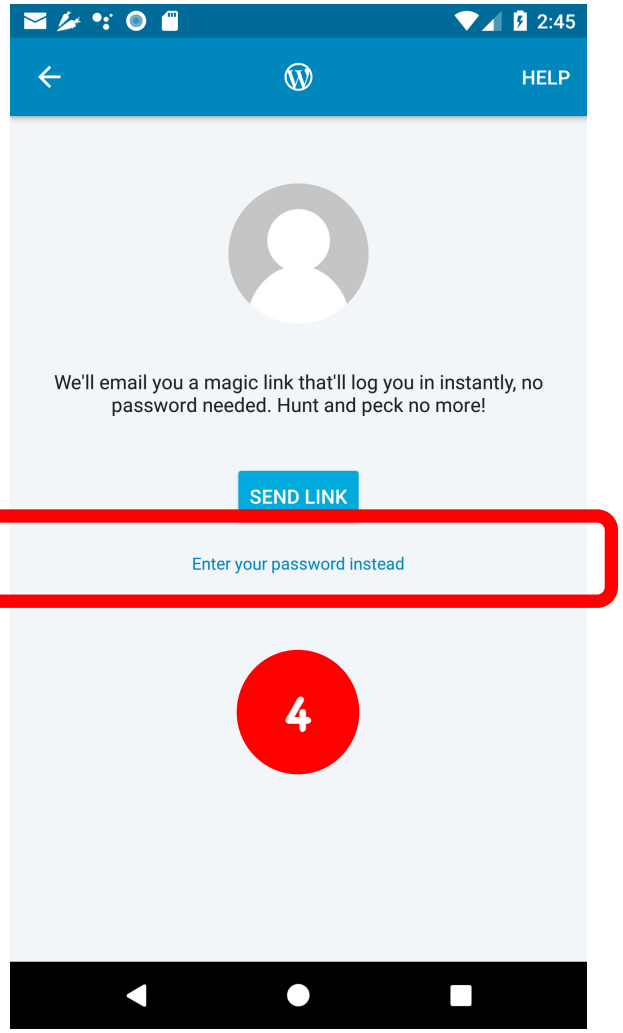
Our Approach

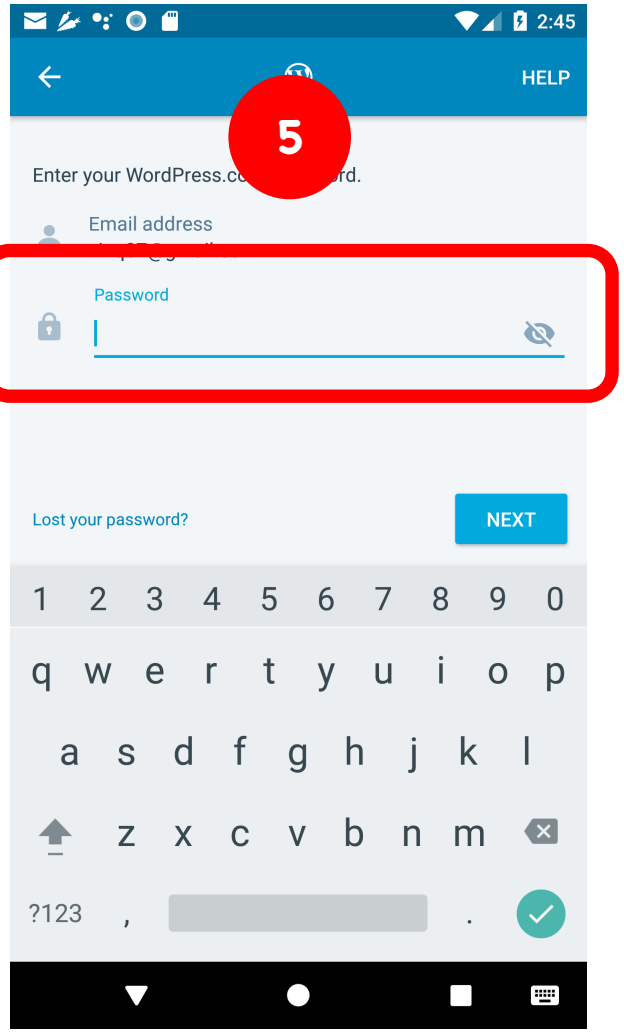
- Identifies features by analyzing execution trace
- Targets Mobile (Android) apps
- In our context, a feature is ***a sequence of user events*** that exercise some functionality of the app

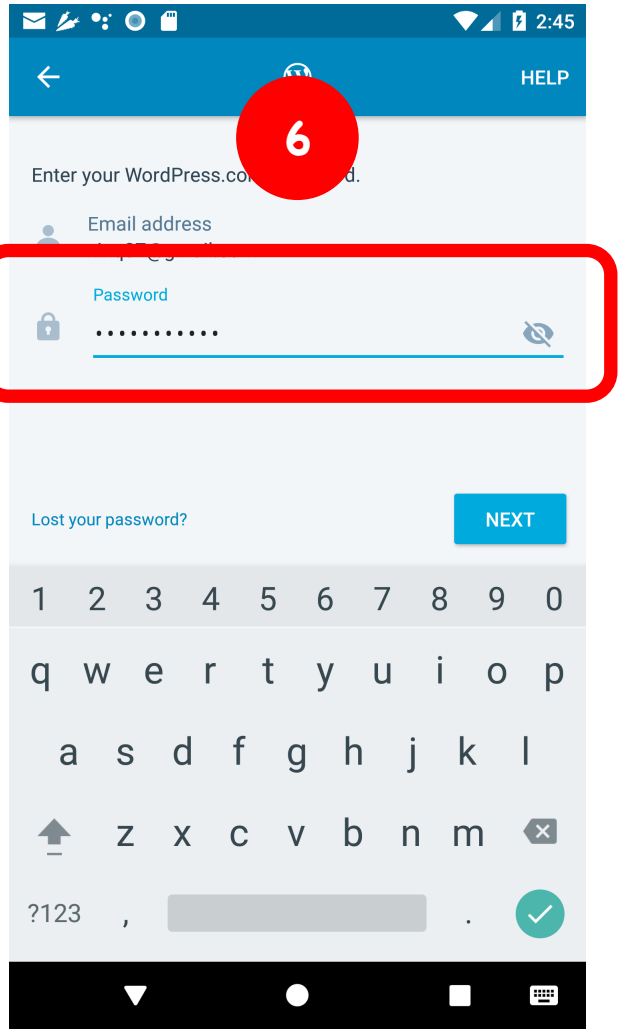
Login Feature of WordPress



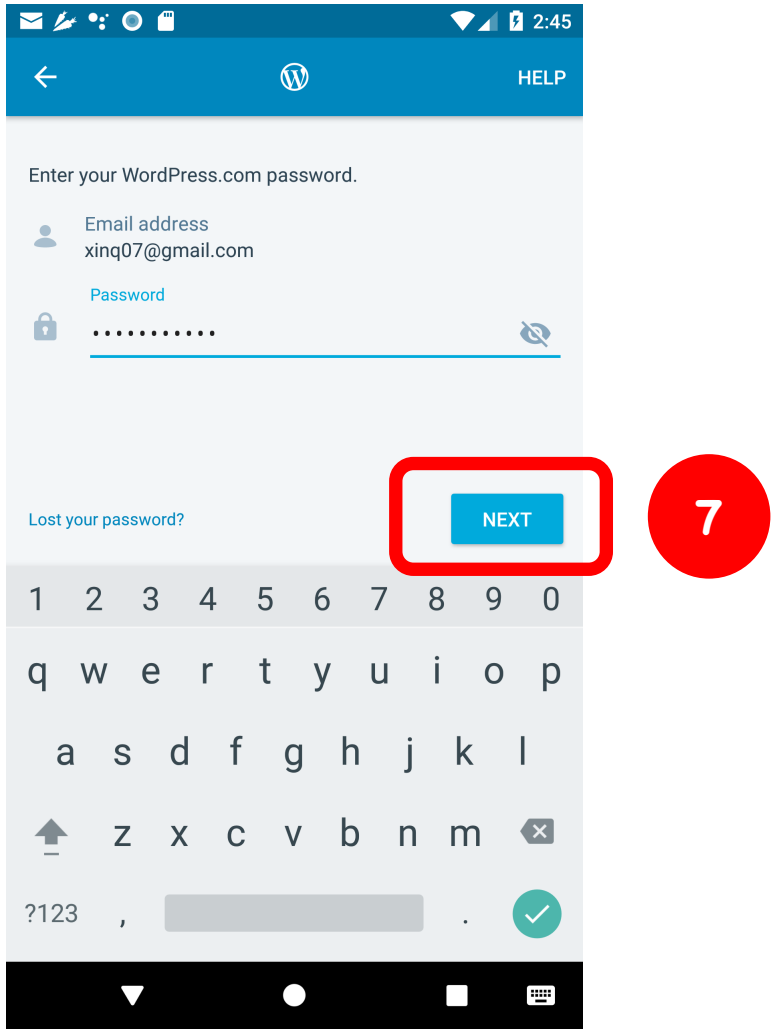
Login Feature of WordPress





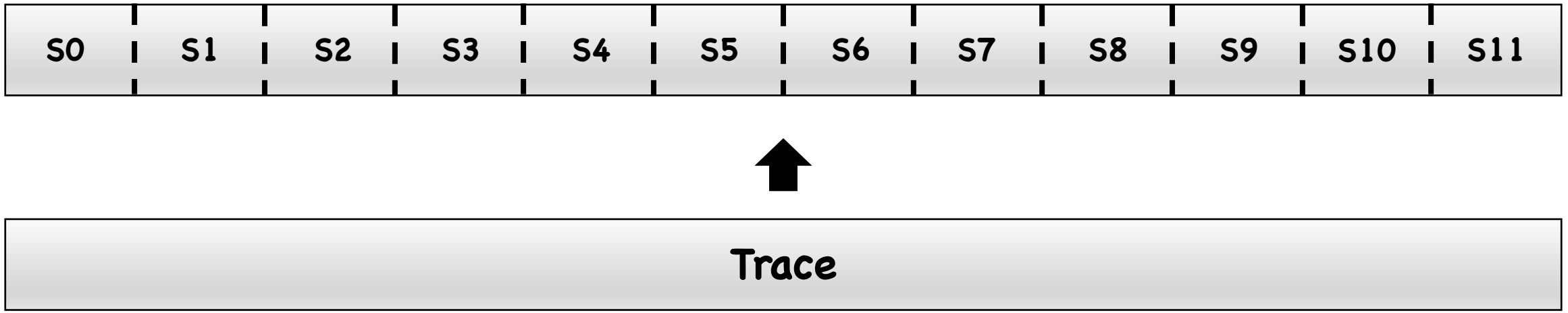


Login Feature of WordPress

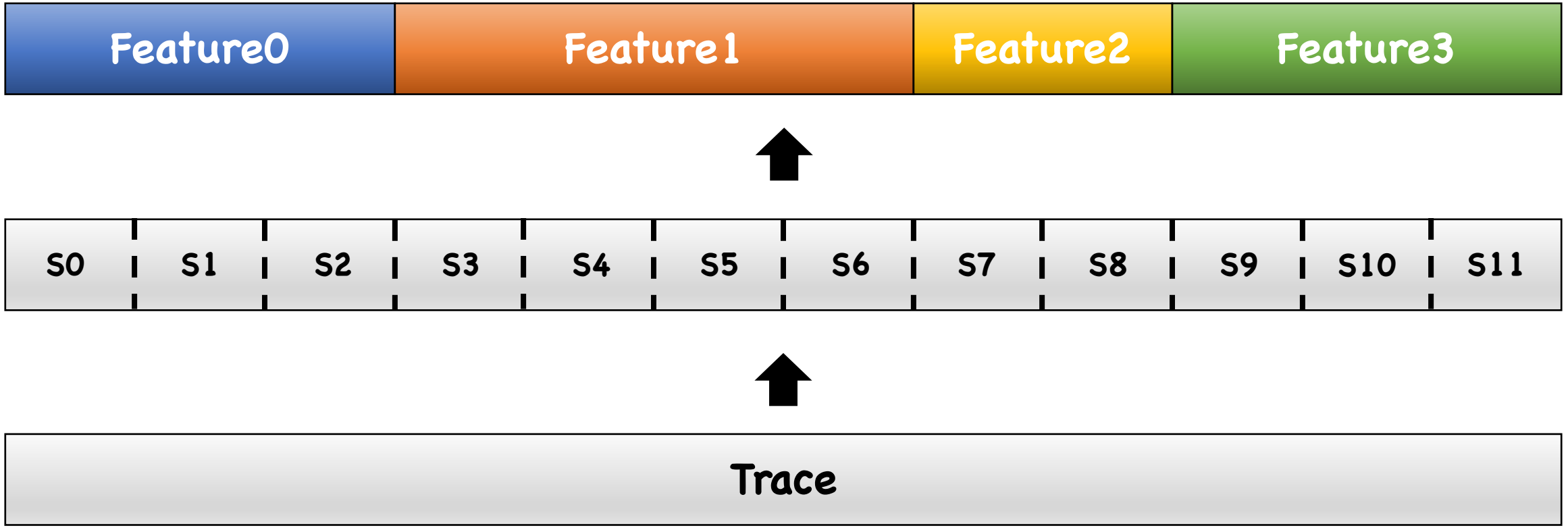


High-level Approach

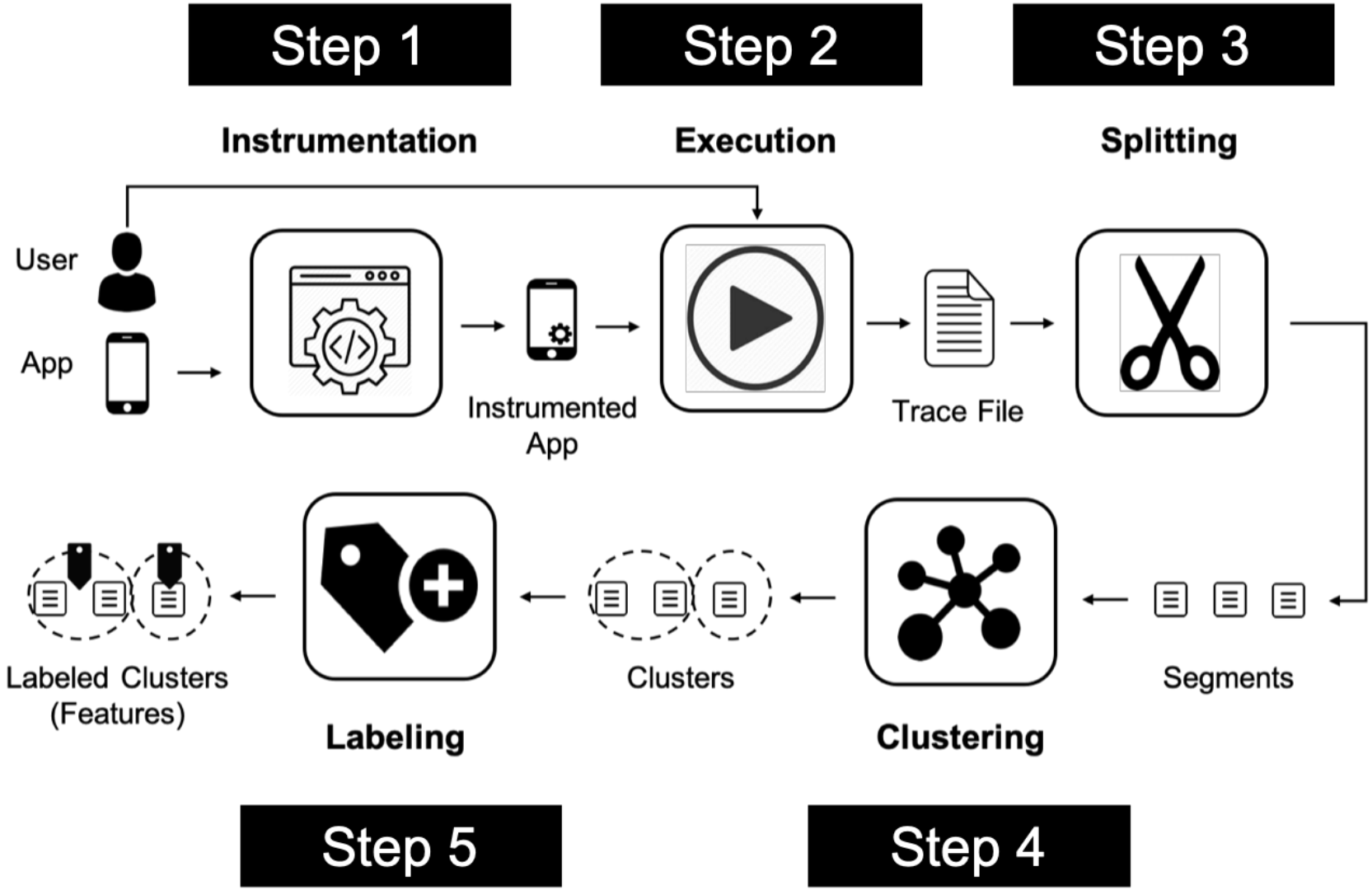
High-level Approach



High-level Approach

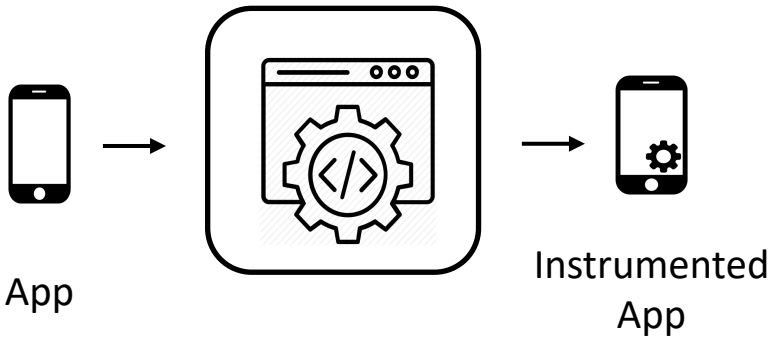


FeatureFinder



FeatureFinder

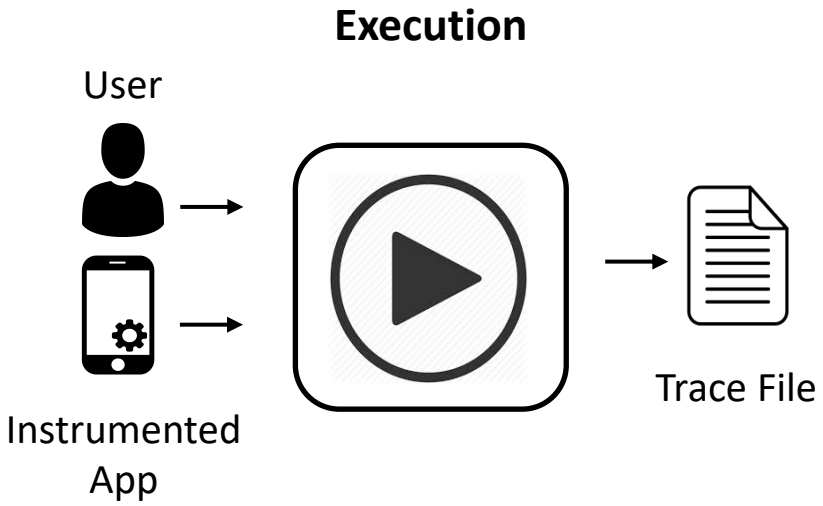
Instrumentation



Step 1

Capture execution information
- Stacks of Method Calls
- Activities & Fragments
- User Events
  - Touch event & widgets
  - Keyboard event & labels

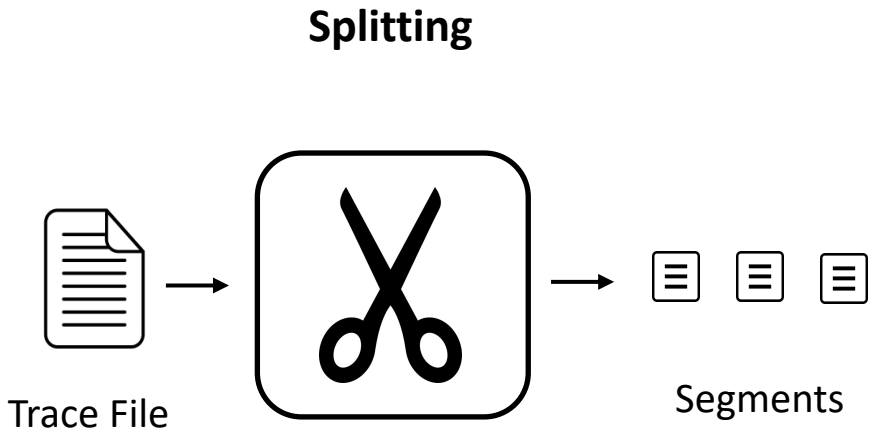
FeatureFinder



User executes the app to exercise its features

Step 2

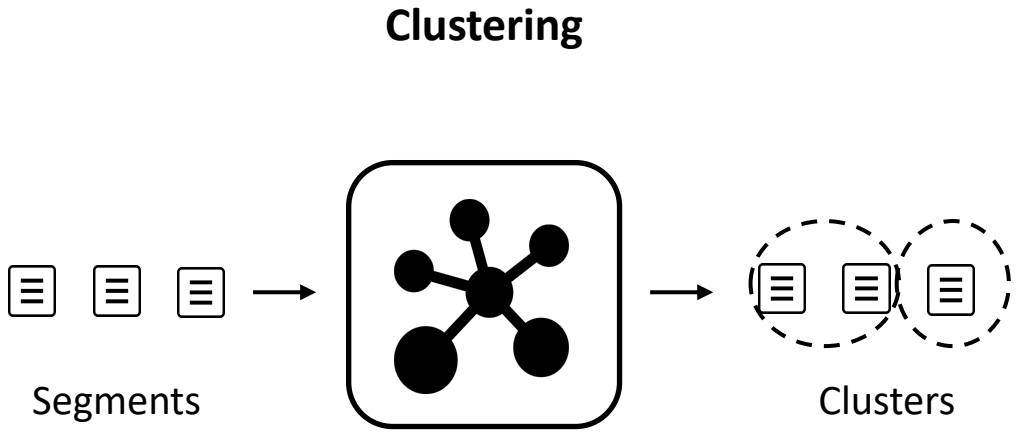
FeatureFinder



Split the trace into segments by user events

Step 3

FeatureFinder

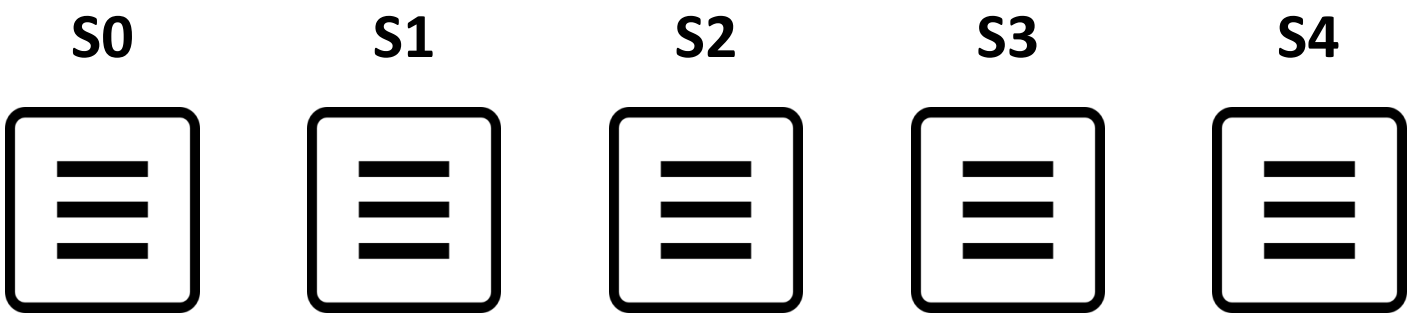


Group "related" segments

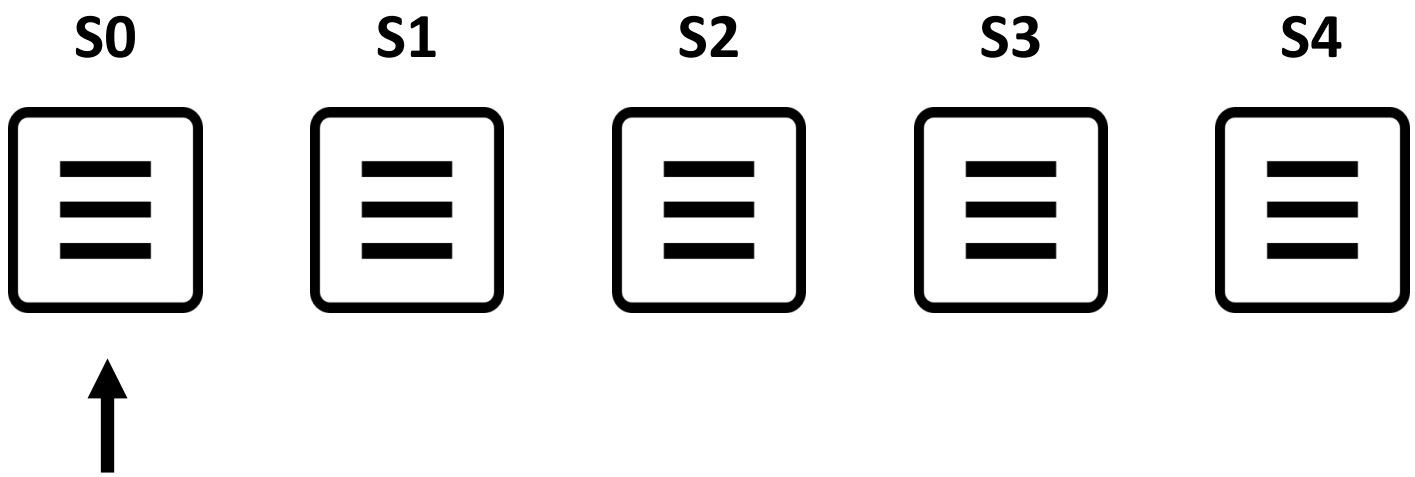
- Compare execution info
- Use a classifier to decide "relatedness"

Step 4

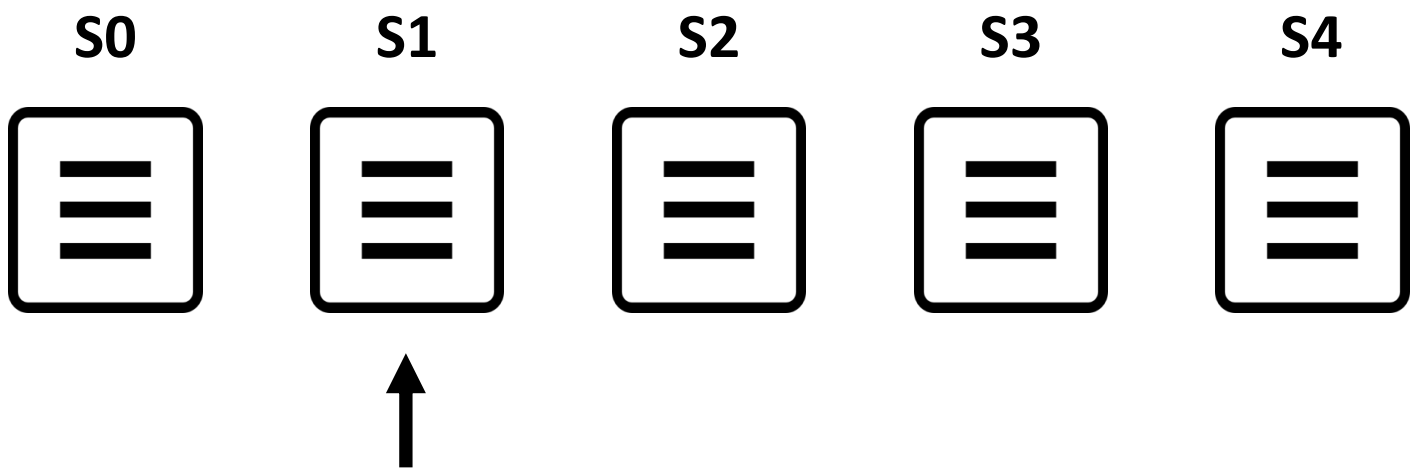
Clustering algorithm



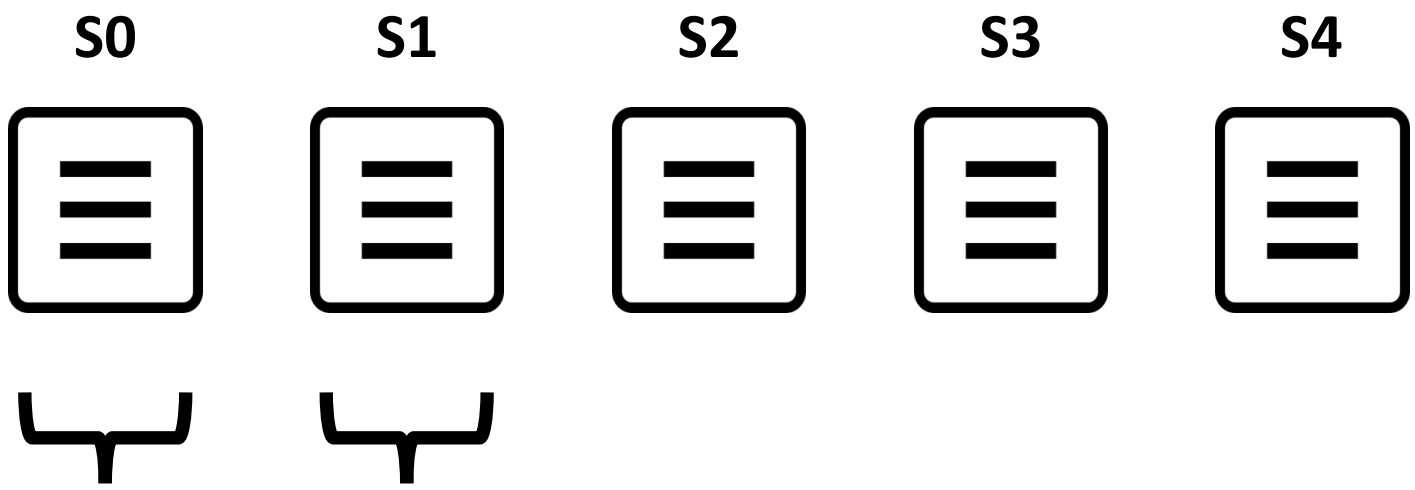
Clustering algorithm



Clustering algorithm



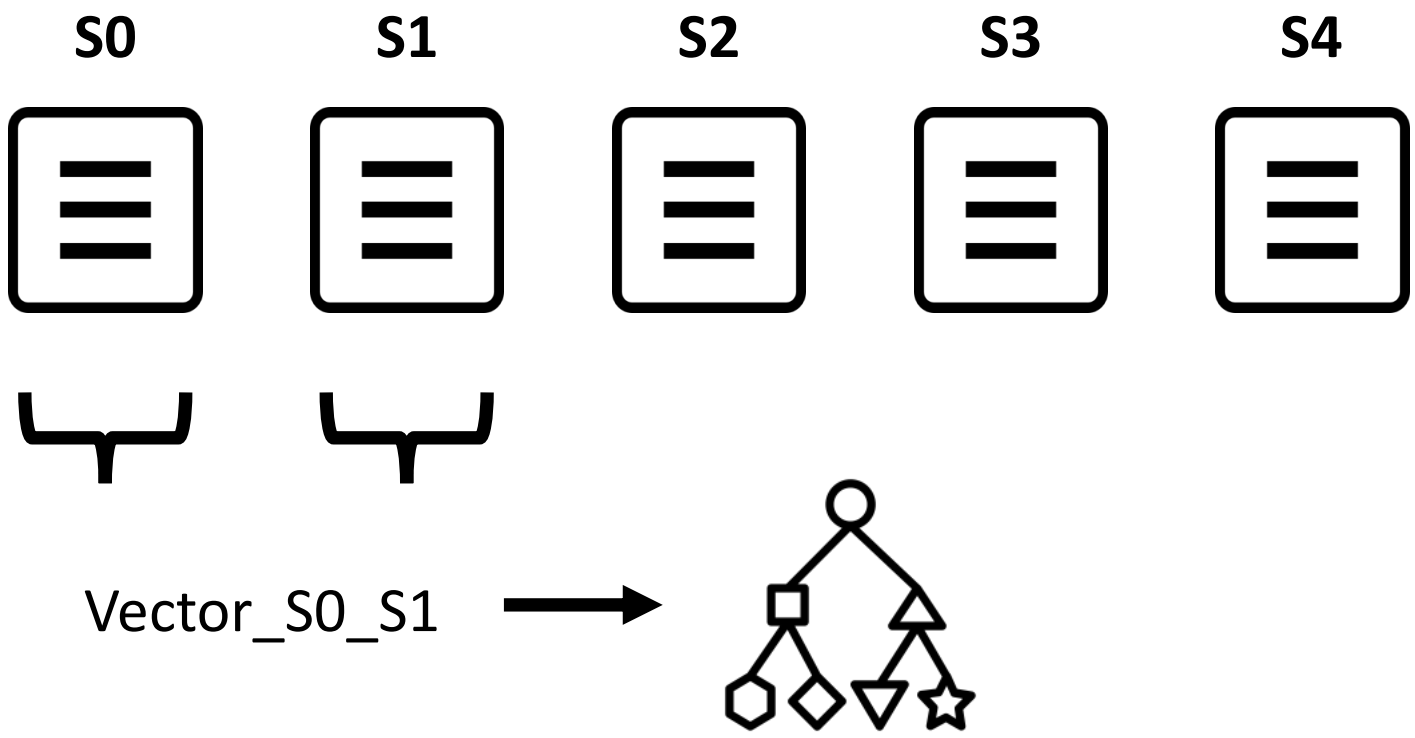
Clustering algorithm



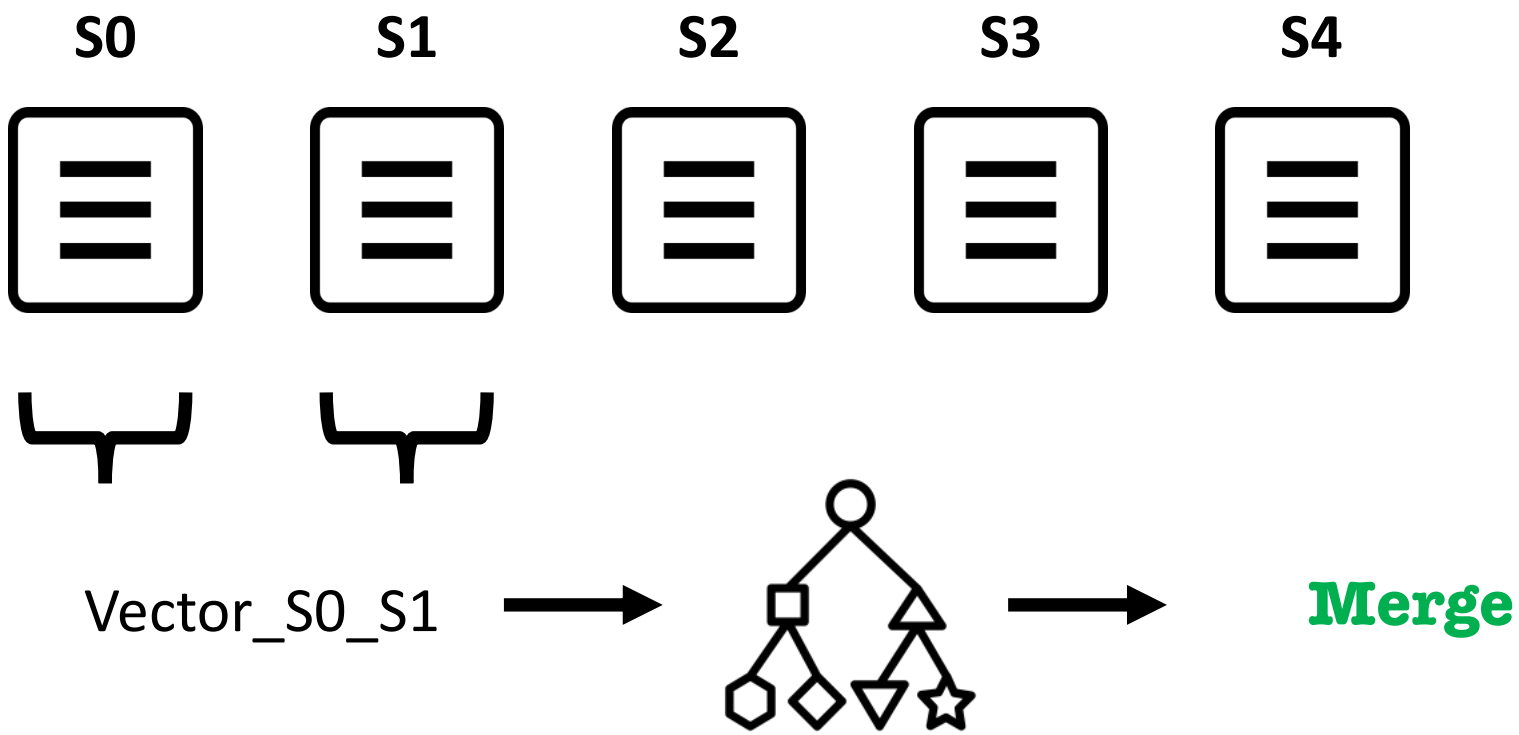
Vector_S0_S1

A numeric vector encoded as the comparison b/w S0 and S1

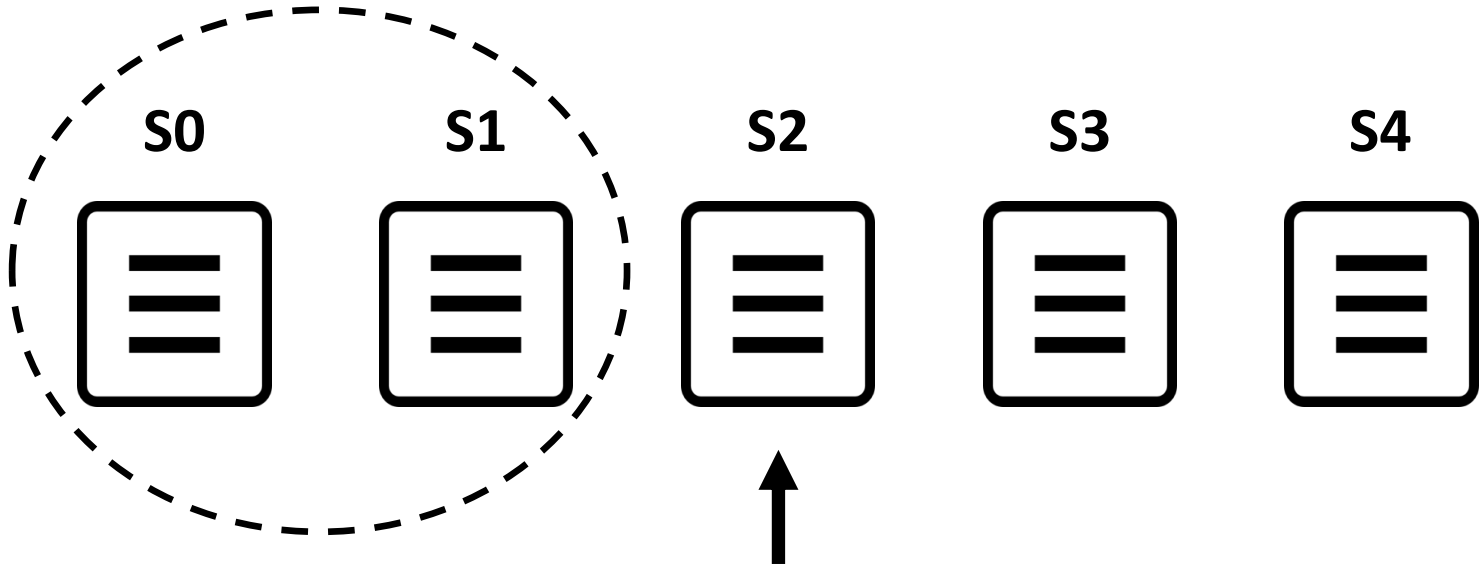
Clustering algorithm



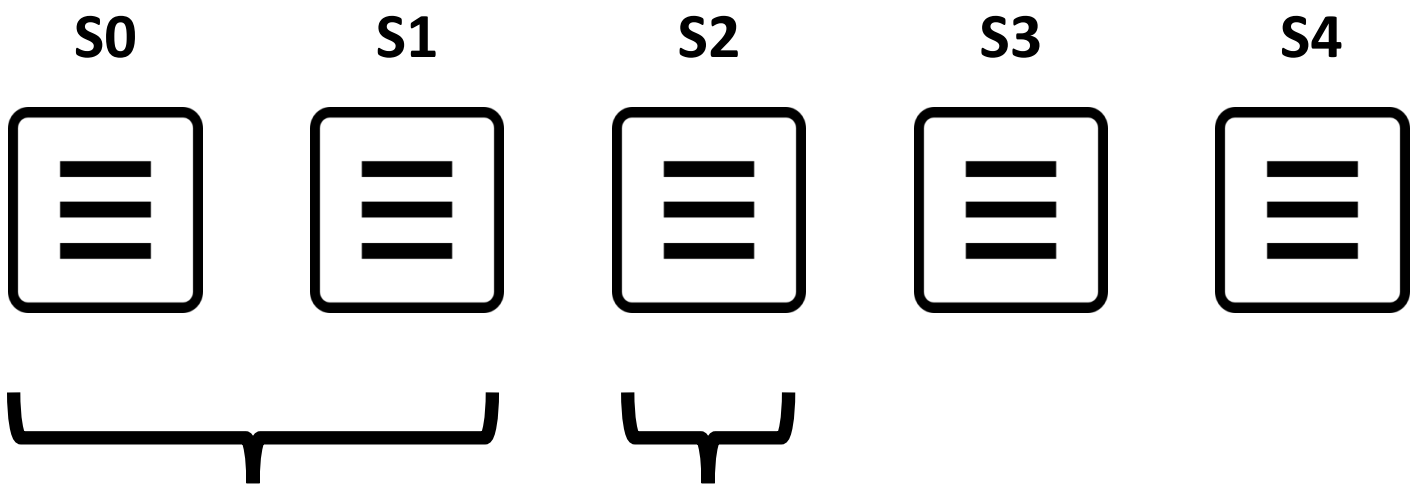
Clustering algorithm



Clustering algorithm

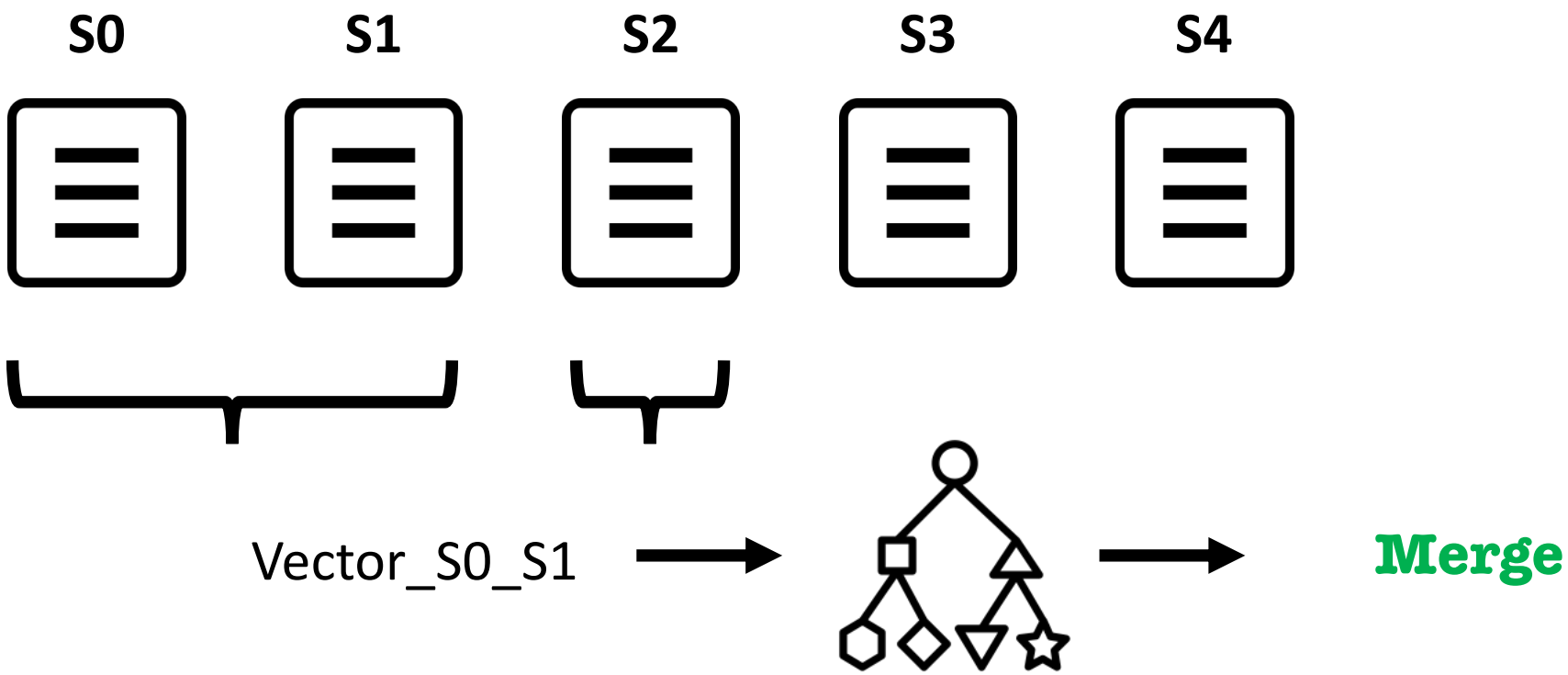


Clustering algorithm

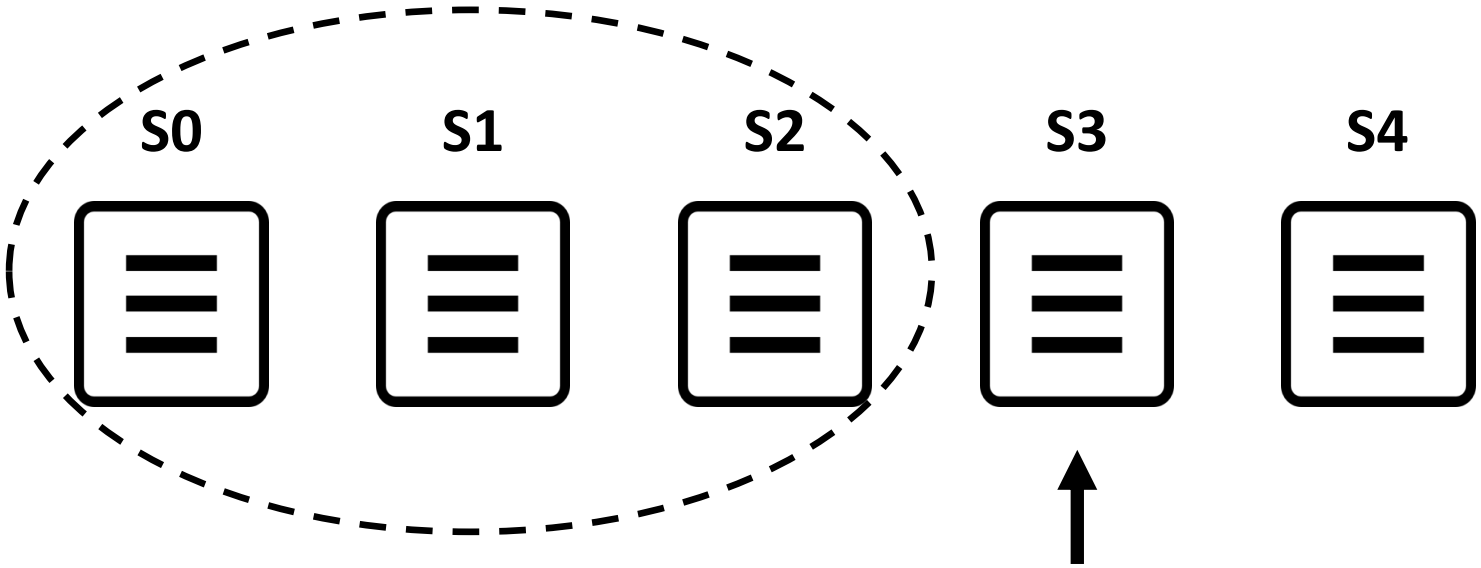


Vector_S0_S1

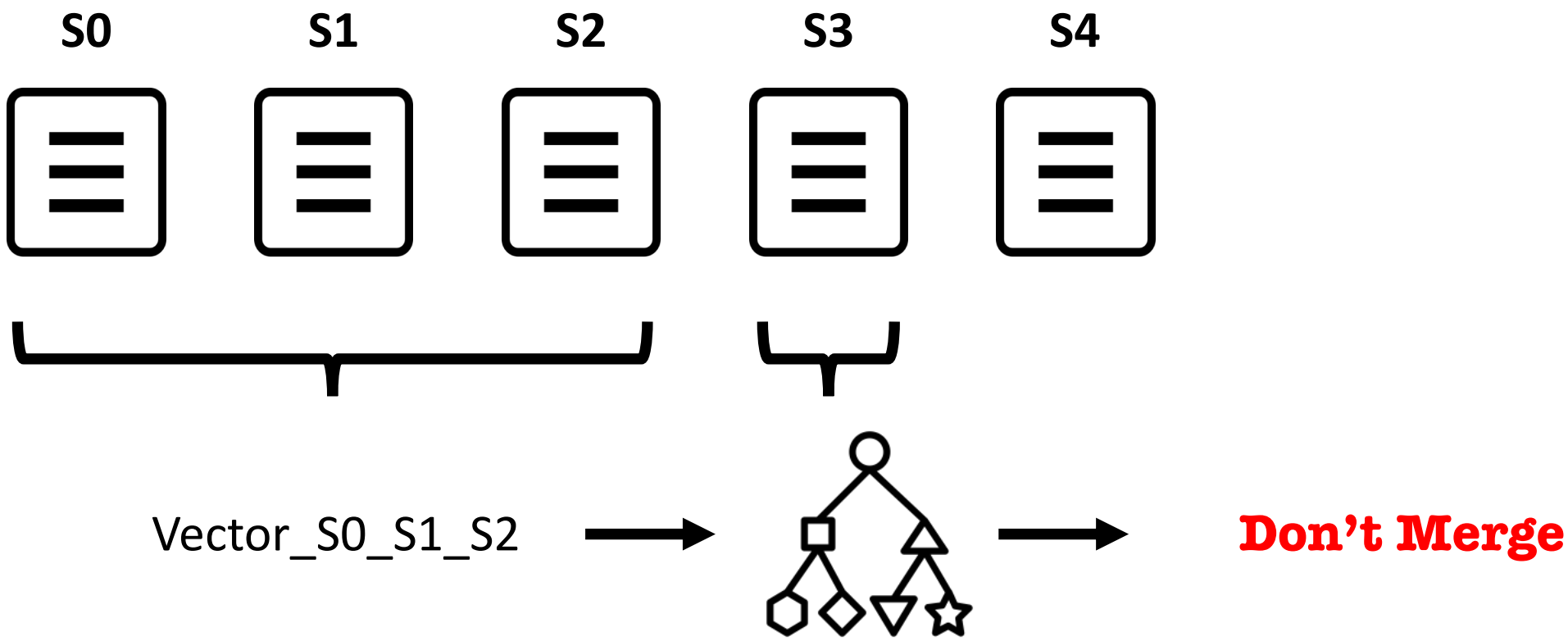
Clustering algorithm



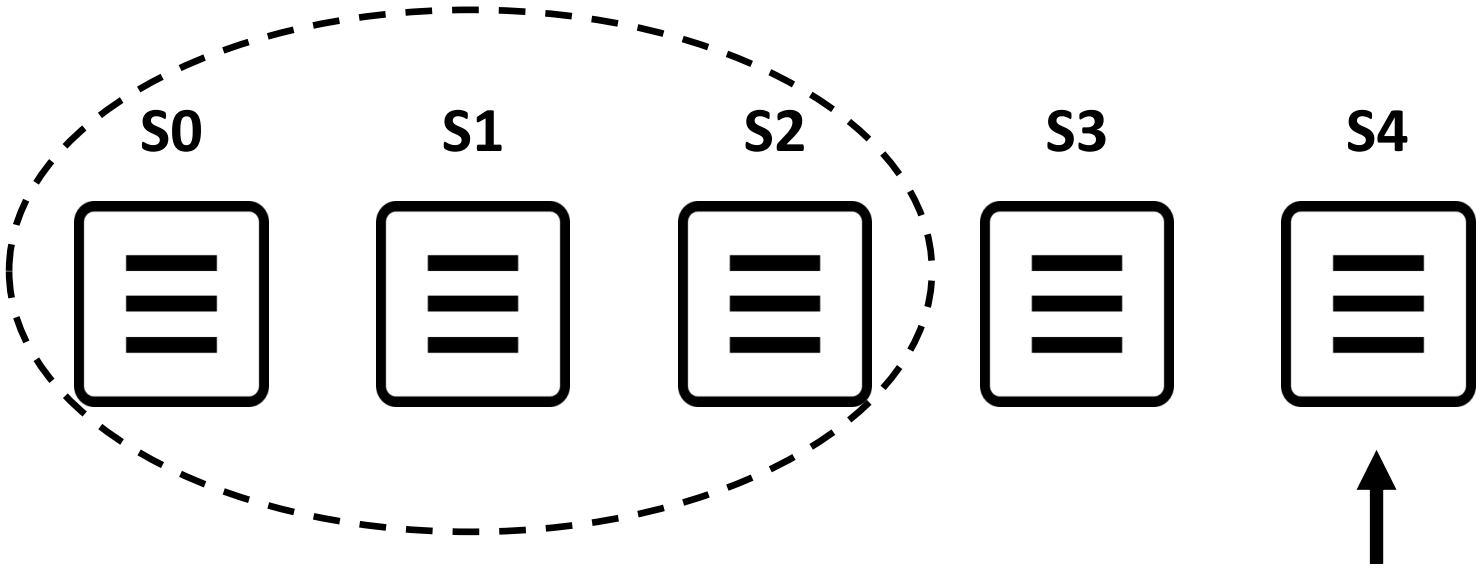
Clustering algorithm



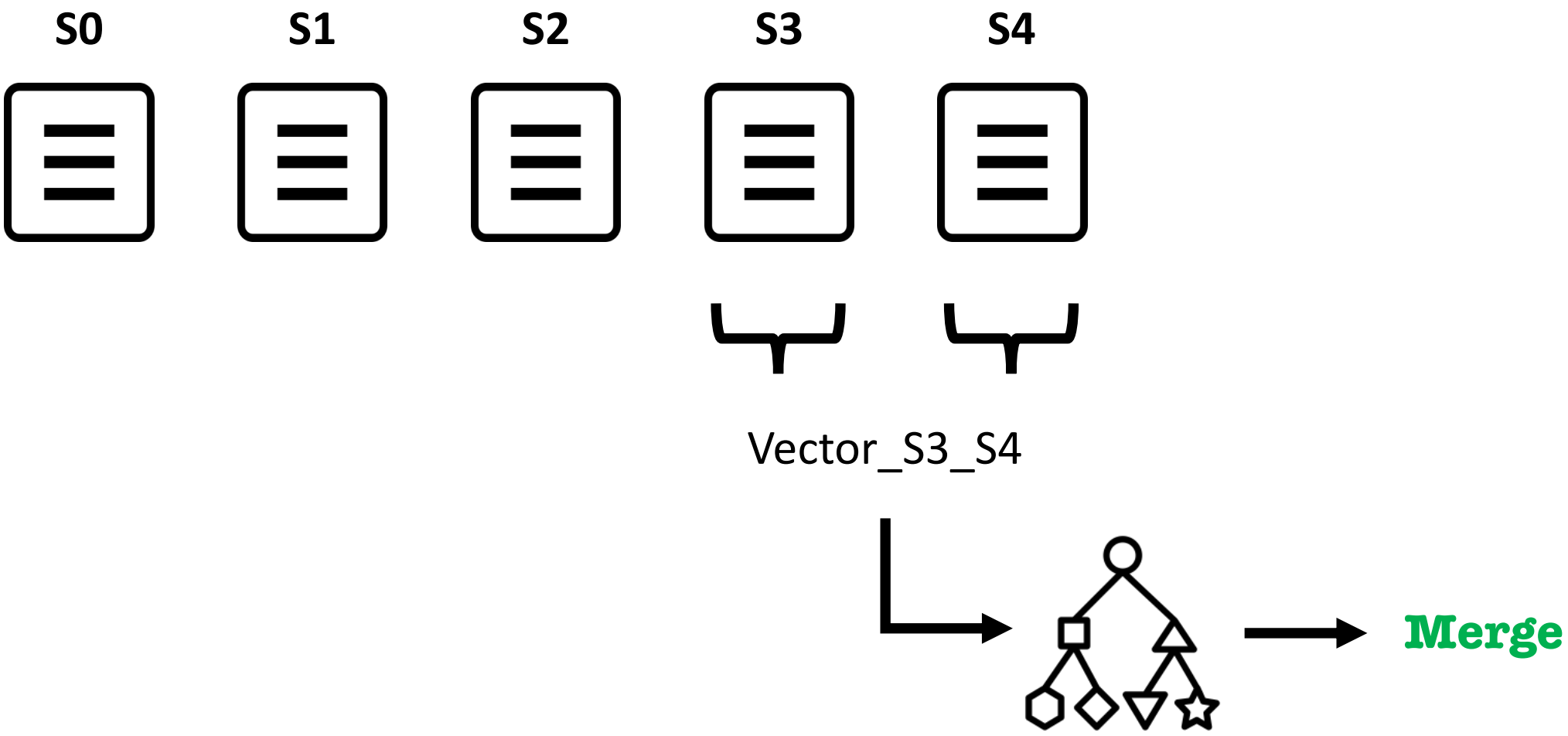
Clustering algorithm



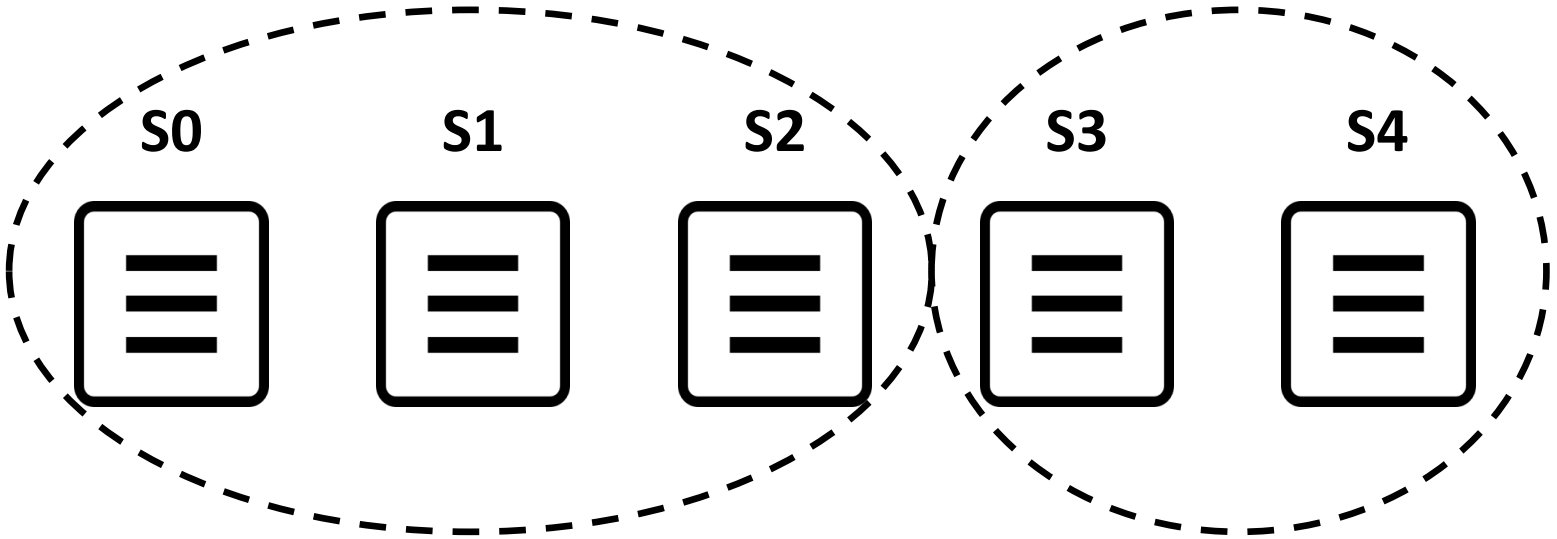
Clustering algorithm



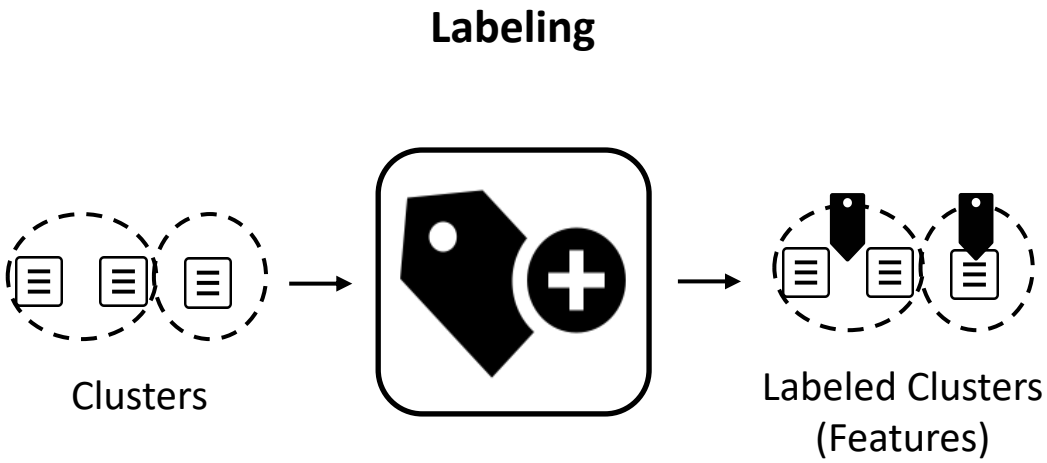
Clustering algorithm



Clustering algorithm



FeatureFinder



Label each cluster

- Use activity & fragment names
- Rank names by TF-IDF values
- Select the top-10

Step 5

Case Study

- Conducted a study using 5 apps
- Exercised different app features and generated traces
- Used ***4 apps*** for classifier training
- Evaluated FeatureFinder on the other app ***K-9 Mail***

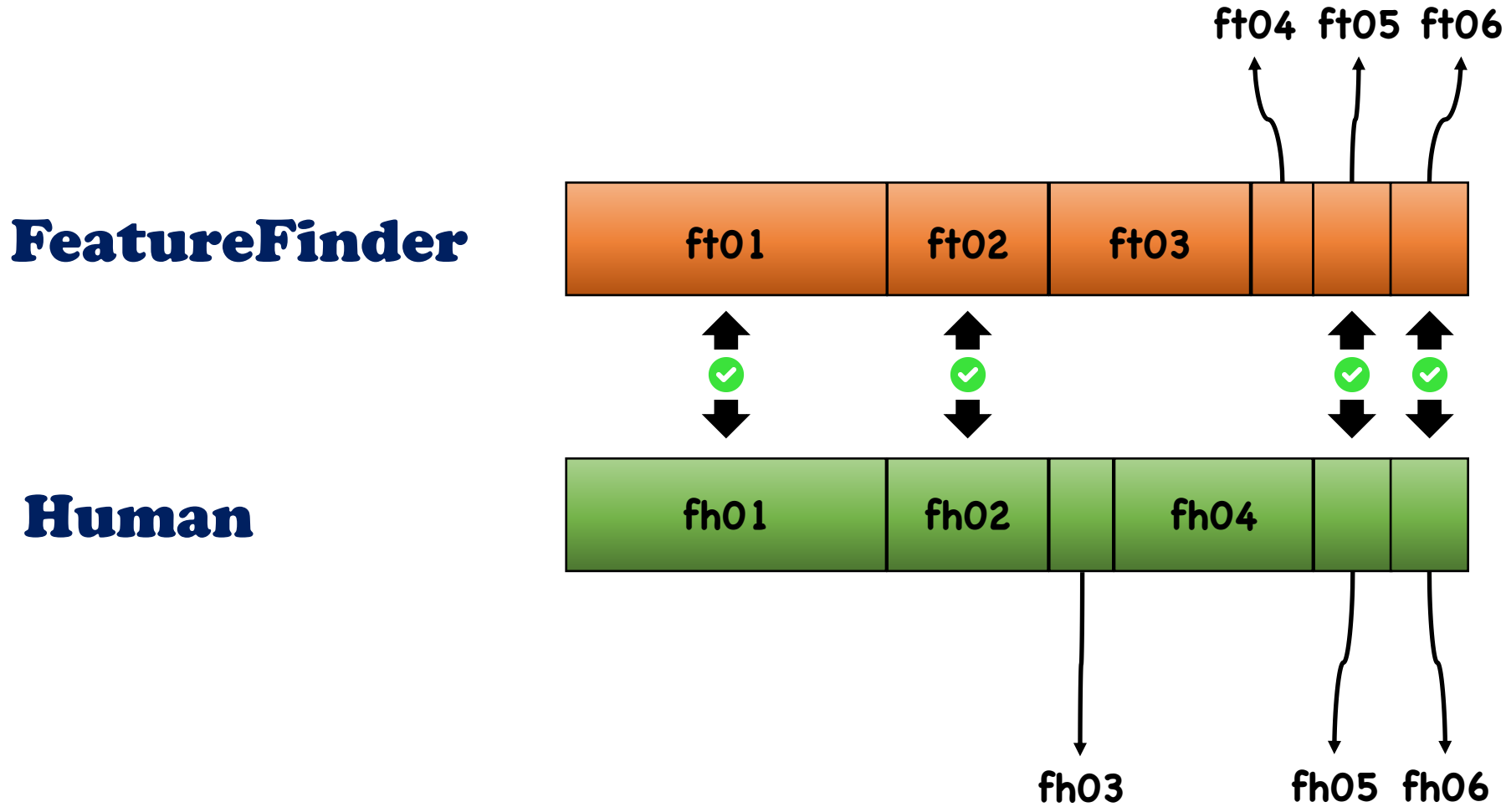
WordPress
DailyMoney
PasswordMaker
Music Player
K-9 Mail

Two Trace for each App

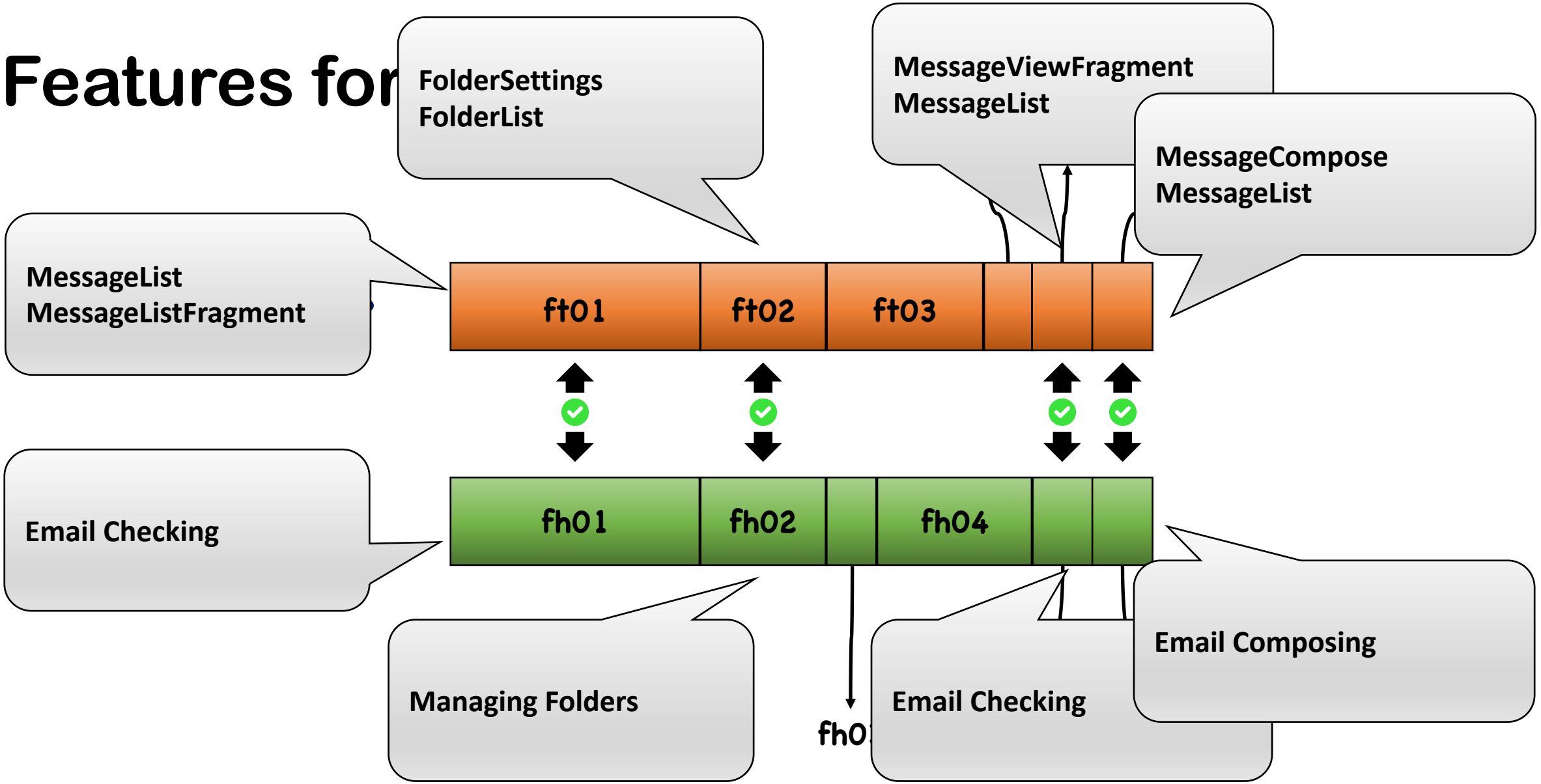
Classifier Training

- Used FeatureFinder to split trace into segments
- Manually Identified clusters
- Generated ***490 segments pairs*** labeled as “**Merge**” & “**Don’t merge**”
- Trained ***10 classifiers***
- Used the best: ***k-NN (k=10)***

Features for Trace 0



Features for



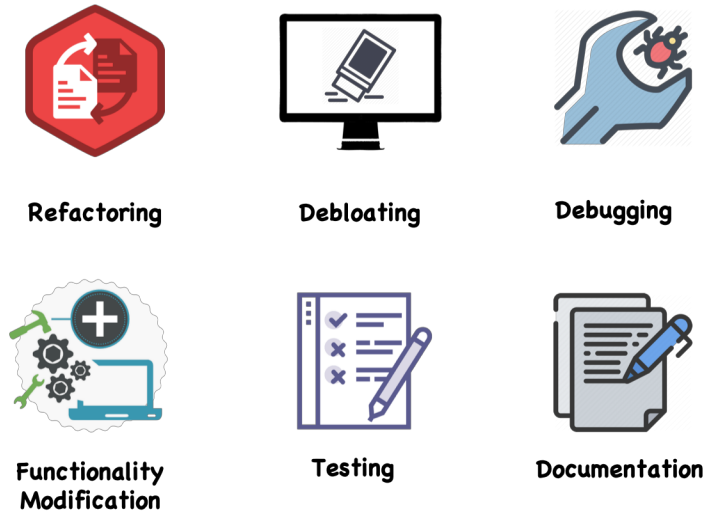
Evaluation Results

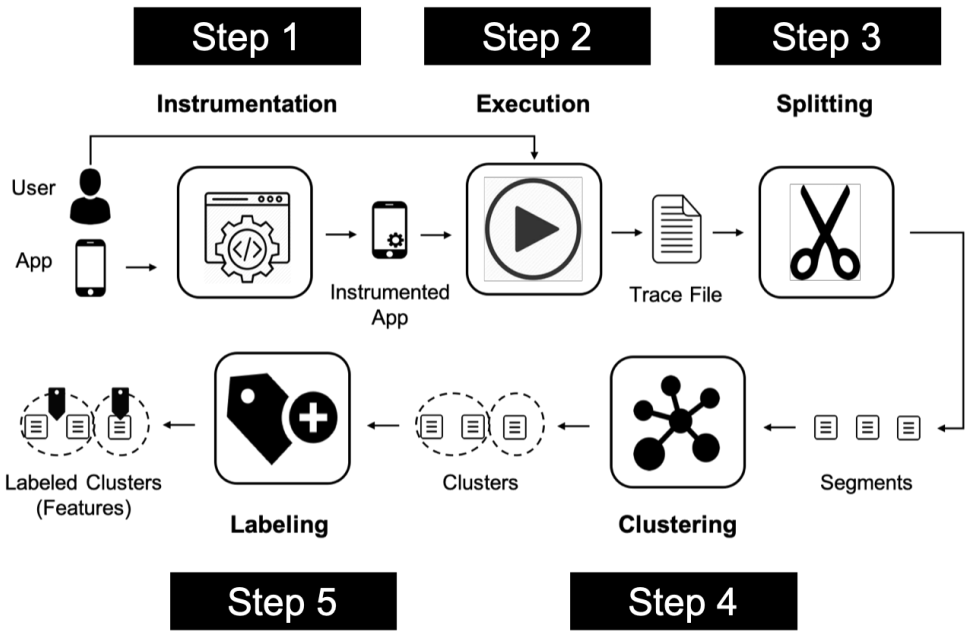
- Manually identified 11 feature clusters (**ground truth**)
- FeatureFinder generated 9 clusters
- Identified 6 of the 11 (***55%***) features
- Labels generated are ***in close meaning*** to the human labels

Conclusion & Future work

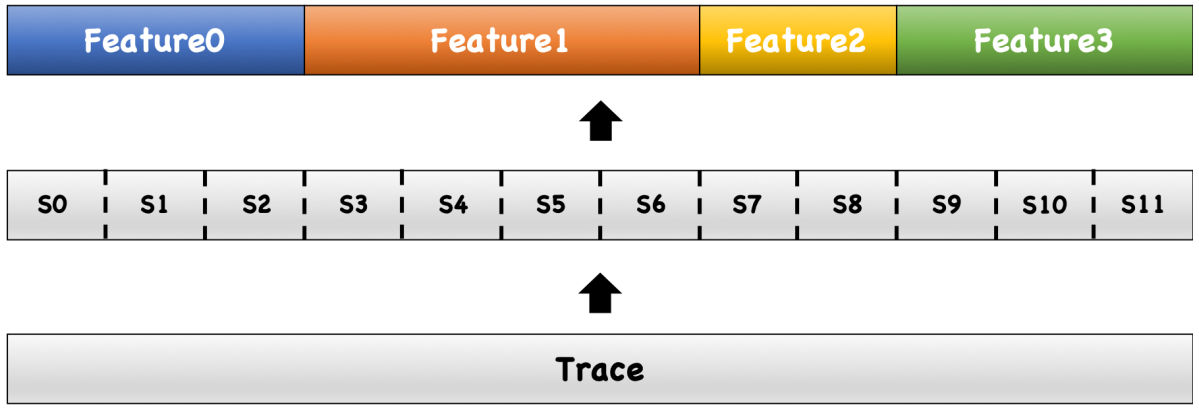
- FeatureFinder identifies ***features*** from app's ***execution traces***
- Case study results, albeit preliminary, are promising
- As future work
  - Perform a user study
  - Extend FeatureFinder to identify features hierarchically
  - Define a visualization for showing the features

Understanding a program & its features





High-level Approach



Evaluation Results

- Manually identified 11 feature clusters (**ground truth**)
- FeatureFinder generated 9 clusters
- Identified 6 of the 11 (***55%***) features
- Labels generated are ***in close meaning*** to the human labels