Demonstrating Praxi

SOFTWARE DISCOVERY THAT LEARNS FROM PRACTICE

BY ANTHONY BYRNE[1], SADIE L. ALLEN[1], SHRIPAD NADGOWDA[2], AND AYSE K. COSKUN[1]

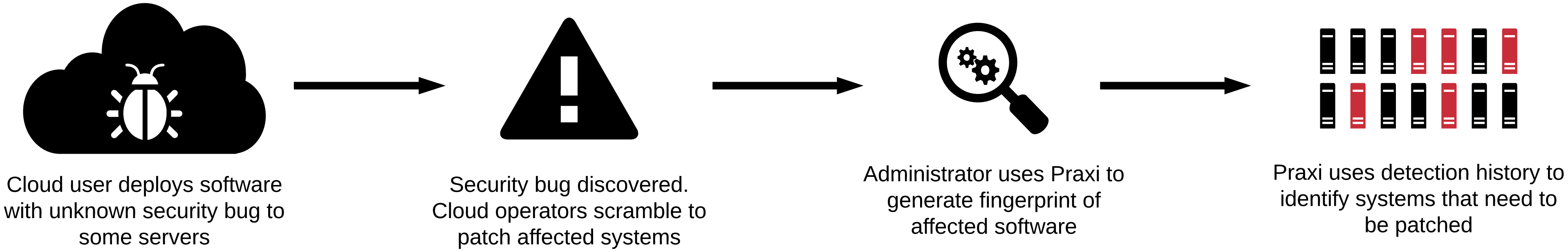
[1]Boston University; [2]IBM T.J. Watson Research Center

20th ACM/IFIP International Middleware Conference
December 11, 2019

BOSTON UNIVERSITY

Motivation

- Cloud systems (bare metal, VMs, containers) evolve rapidly over time
- New software and vulnerabilities announced every day
- Without constant visibility, cloud software quickly ages/becomes insecure
  - *How do we keep track of what software in installed on a cloud system?*



IBM Research Team

Previous Solution: Statistical Analysis

Columbus: Practice-Based Discovery Method

- Exploit software naming conventions to build modified trie
- Trie then analyzed via freq. counts to pull out significant tags
- Tags hold useful information like app name, version, etc.
- Upside: corpus-less, lightweight
- Downside: tags not consistent/machine-readable

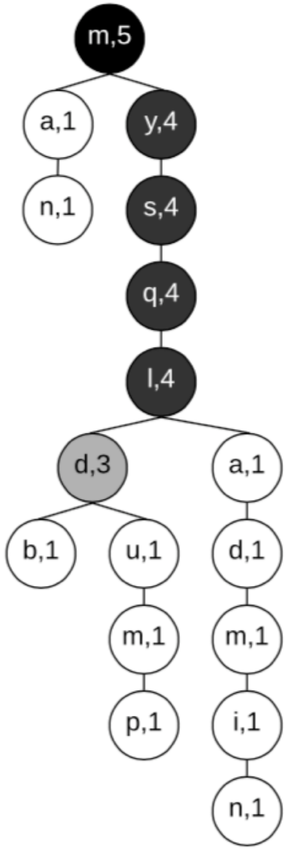


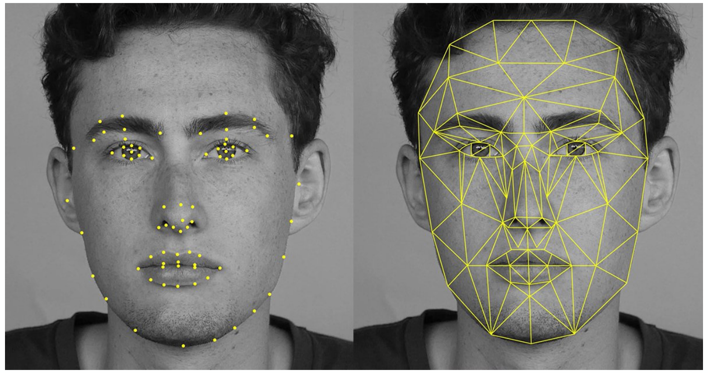
Fig. 1. A frequency trie for the inputs `[man, mysqld, mysqldb, mysqldump, mysqladmin]`. The *non-trivial* tag with the highest frequency is `mysql`, followed by `mysqld`.

S. Nadgowda et al., "Columbus: Filesystem Tree Introspection for Software Discovery" (*IEEE IC2E* 2017)

Key Insight: Discovery By Example

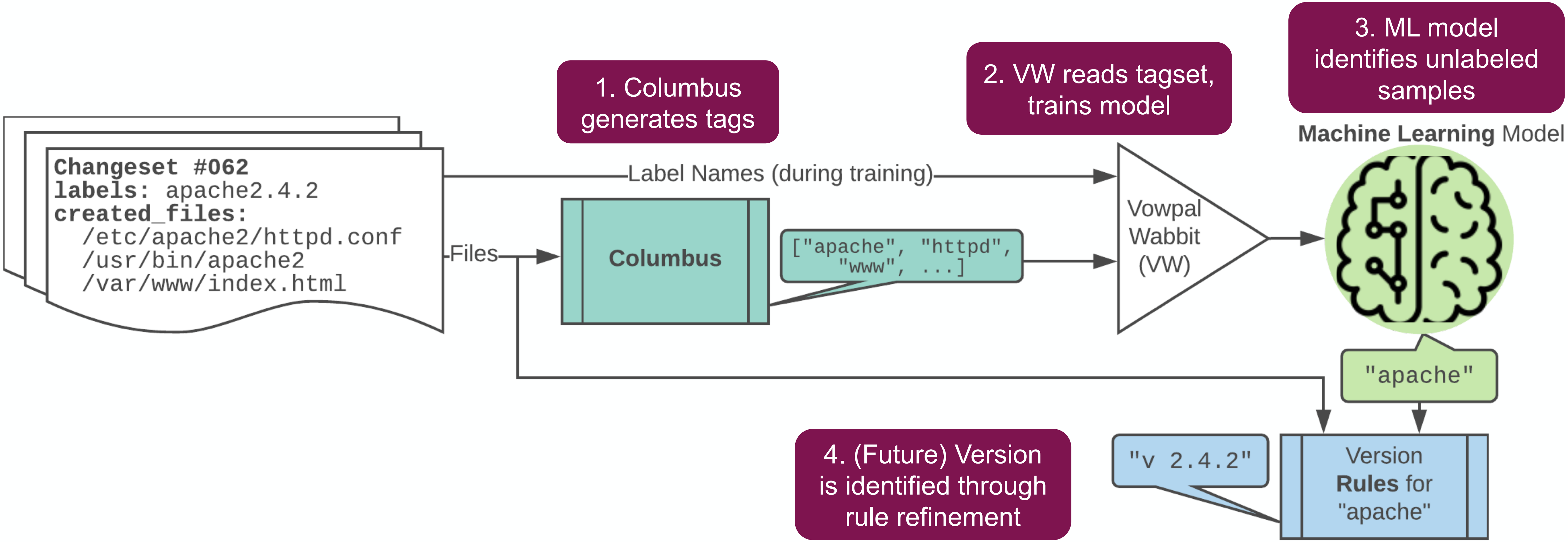
- Machine Learning by Experience
  - Automatic
  - Incremental
  - Generic
  - Distortion resistant

- So how do we apply this to software discovery?



H. Chen et al., "Automated system change discovery and management in the cloud" (*IBM Journ. of R. & D.* 2016)

A. Byrne et al., “Praxi: Cloud Software Discovery That Learns From Practice” (*IEEE TCC*, submitted Nov. ‘18, revised Jul. ‘19)



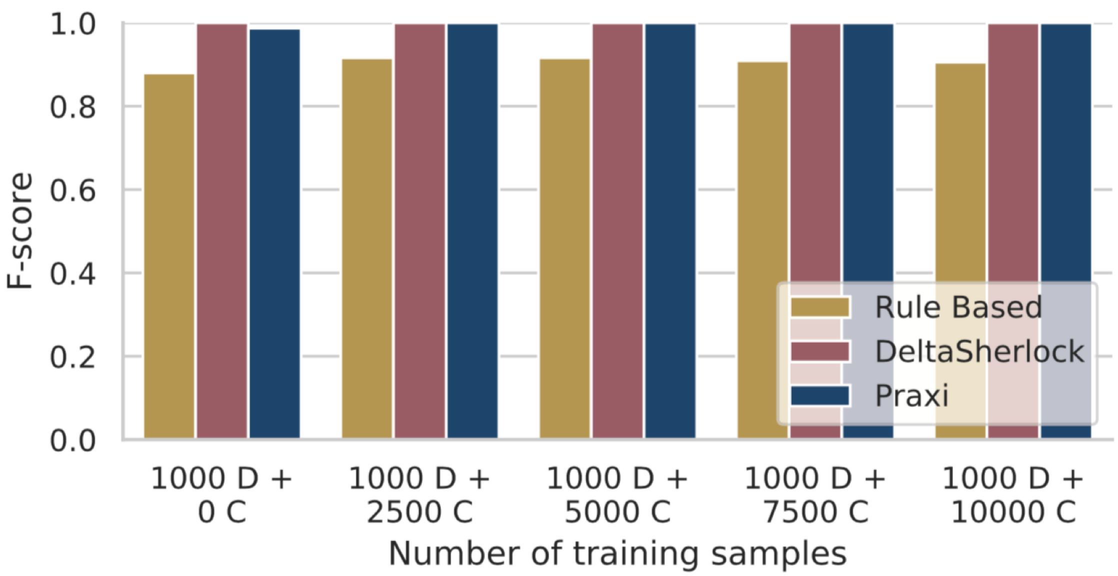
Praxi: Learning From Practice

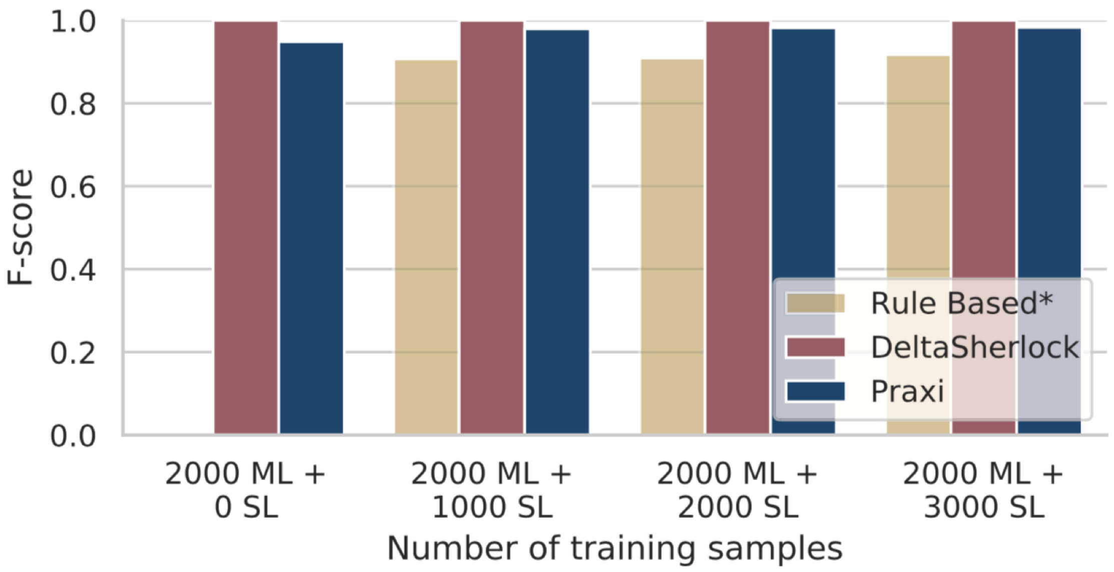
COMBINING THE BEST ELEMENTS OF LEARNING- AND PRACTICE-BASED METHODS

Also seen in: *IEEE IC2E 2019 Tutorials*

Accuracy

(higher F1 scores are better)





Single-label Classification

- Installed one application per recording period
- Average F1 > 0.99

Multi-label Classification

- Installed multiple applications per recording period
- Average F1 = 0.967

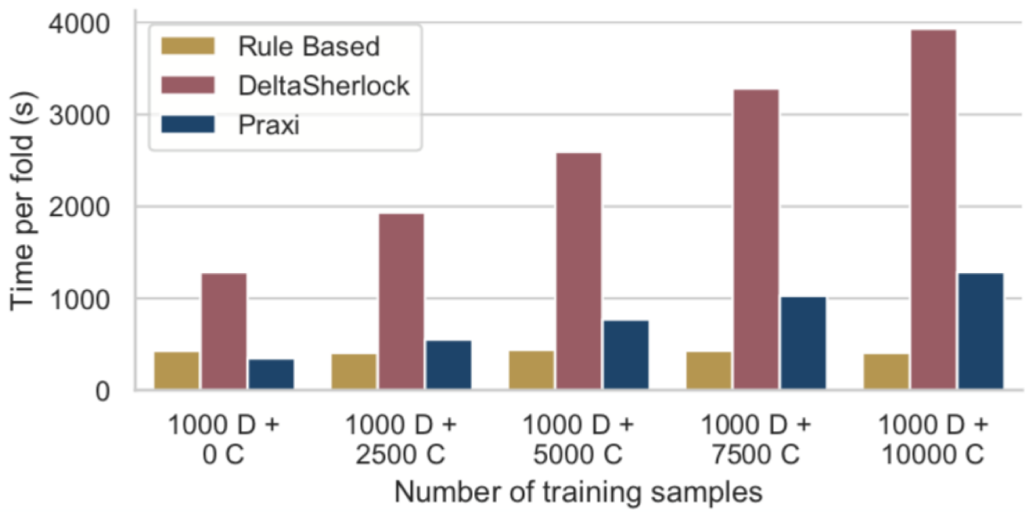
Praxi Overhead Compared to Previous Work

COMPARISON OF OVERALL OVERHEAD FOR MULTI-LABEL CLASSIFICATION

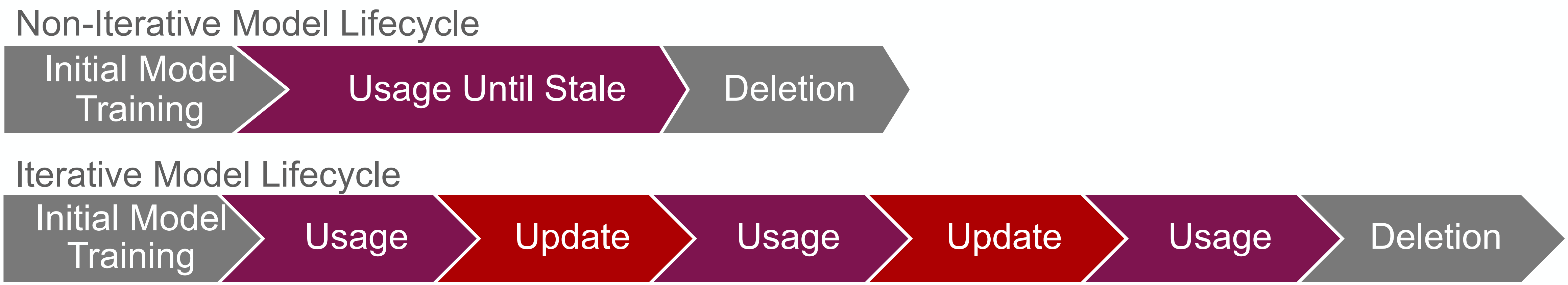
	Praxi			DeltaSherlock		
Phase	**Operation**	**Time (min)**	**Disk (MB)**	**Operation**	**Time (min)**	**Disk (MB)**
Feature Reduction	Columbus Tag Extraction	3.7	55	w2v Dictionary Generation	13.1	370
				Fingerprinting	55	24
Discovery By Example	VW Model Training	1.5	59	RBF Model Training	11	489
	VW Model Evaluation	0.2	-	RBF Model Evaluation	0.7	-
	Overall	**5.4**	**114**	**Overall**	**79.8**	**883**

- Main savings come from...
  - Lack of dictionary generation step
  - Faster machine learning system
  - Smaller machine learning models

Single-label Runtime Comparison

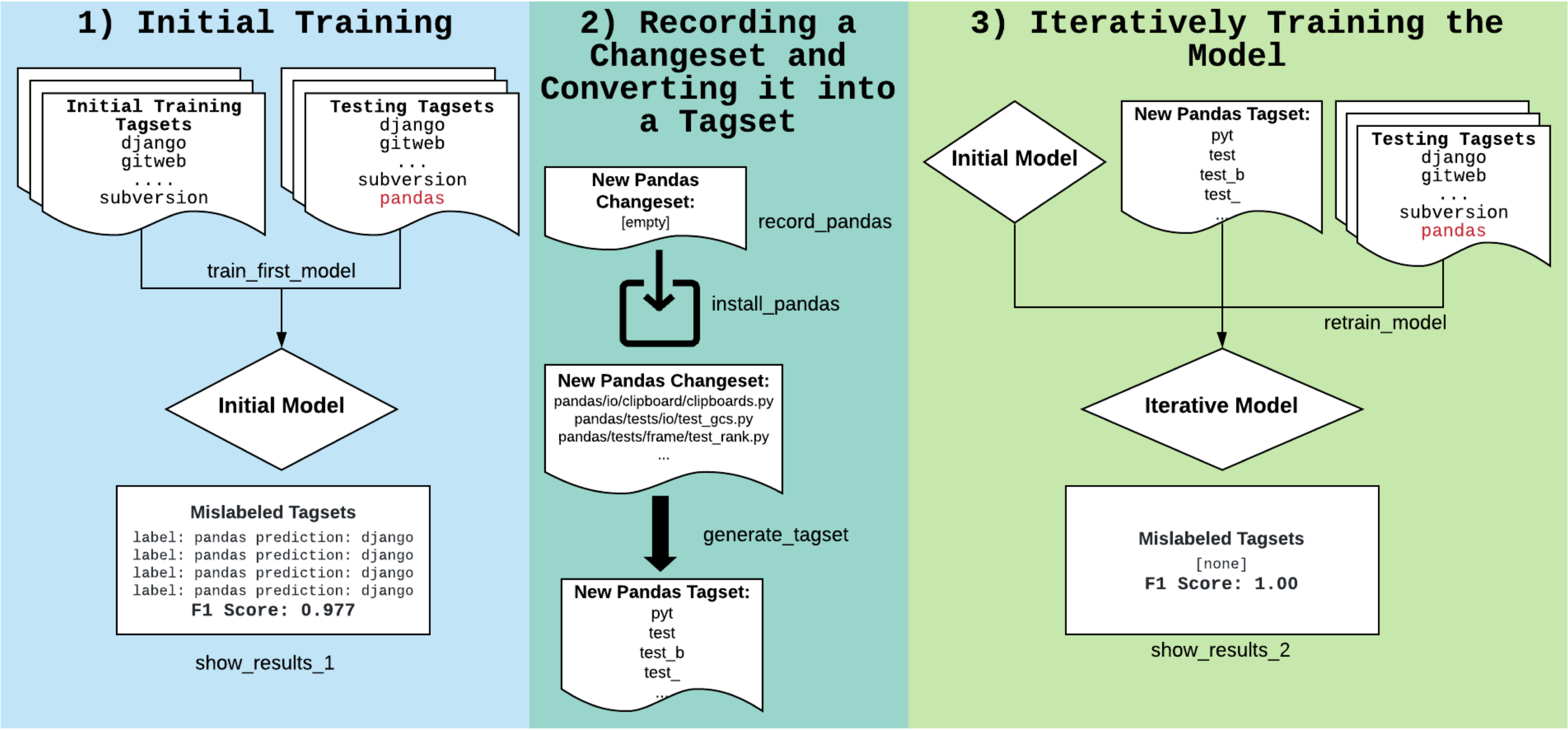


Iterative Training



- Initial ML model training is costly
- Non-iterative models will quickly become ”stale,” requiring full retraining
- Iterative models can be updated several times, minimizing training costs

Live Demo Overview



LiveSlides web content

To view

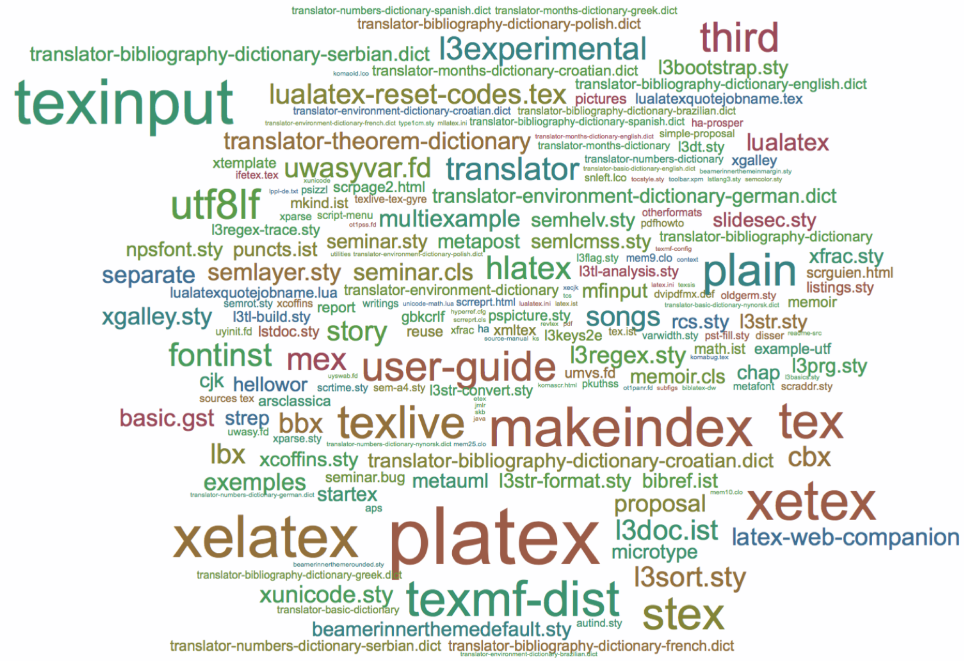
Download the add-in.

liveslides.com/download

Start the presentation.

PEACLAB IBM Research Team BOSTON UNIVERSITY

- Software discovery key to any cloud integrity solution
- Praxi discovers software accurately and automatically with low overhead



Concluding Remarks

More info at bu.edu/peaclab

Please send feedback to abyrne19@bu.edu