

Automated Extraction and Analysis of Developer's Rationale in Open Source Software

SEMTL Meeting @ UdeM



22/09/2025

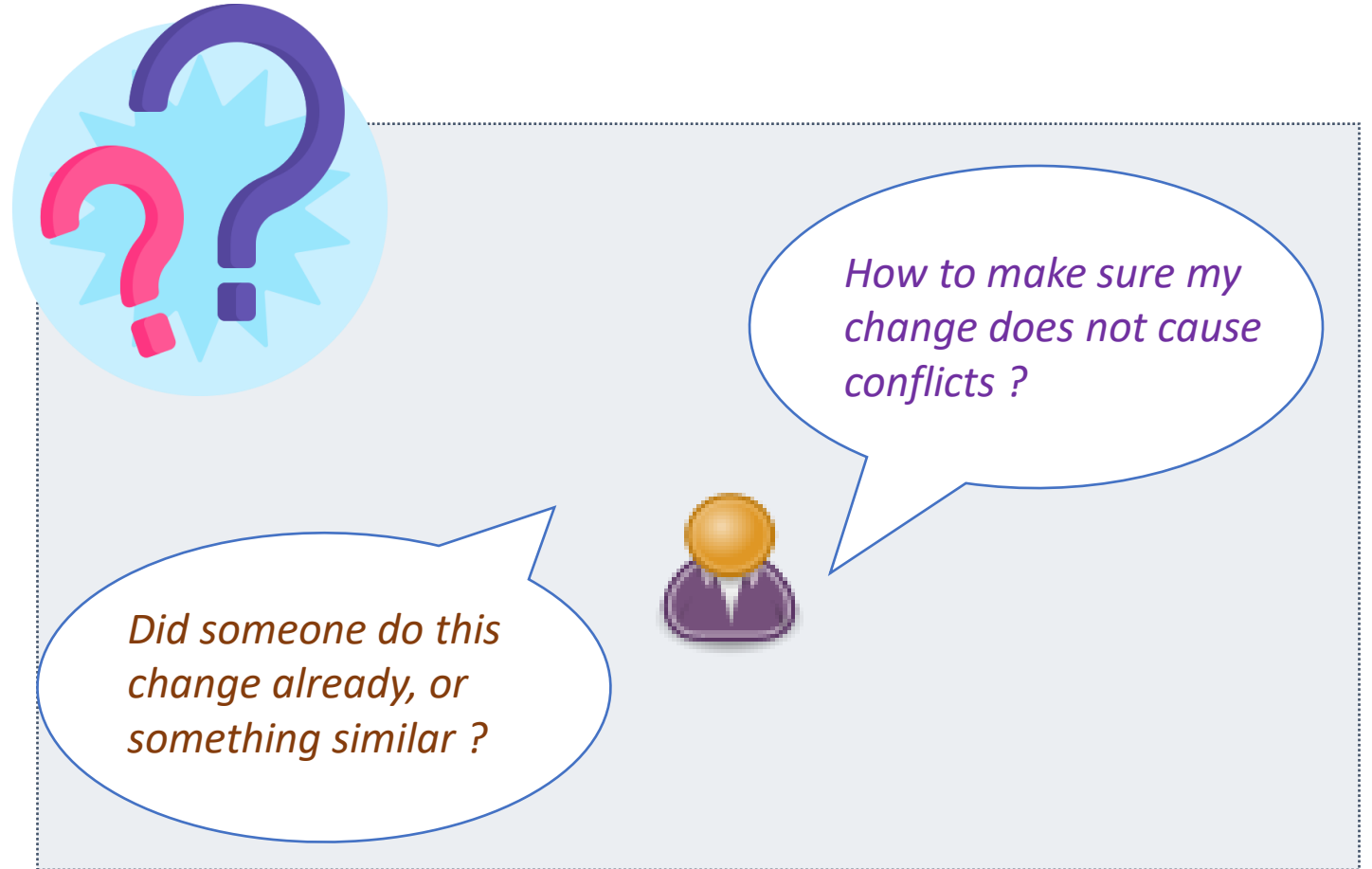
Mouna Dhaouadi, Bentley Oakes and Michalis Famelis

PACMSE'25

Two challenges

How can we exploit Rationale Information to:

- Reuse past solutions 
- Avoid reasoning conflicts 



Commit Messages as Data Source



- *Commit Messages* contain valuable information about developers rationale
- Rationale is the **why** behind decisions.

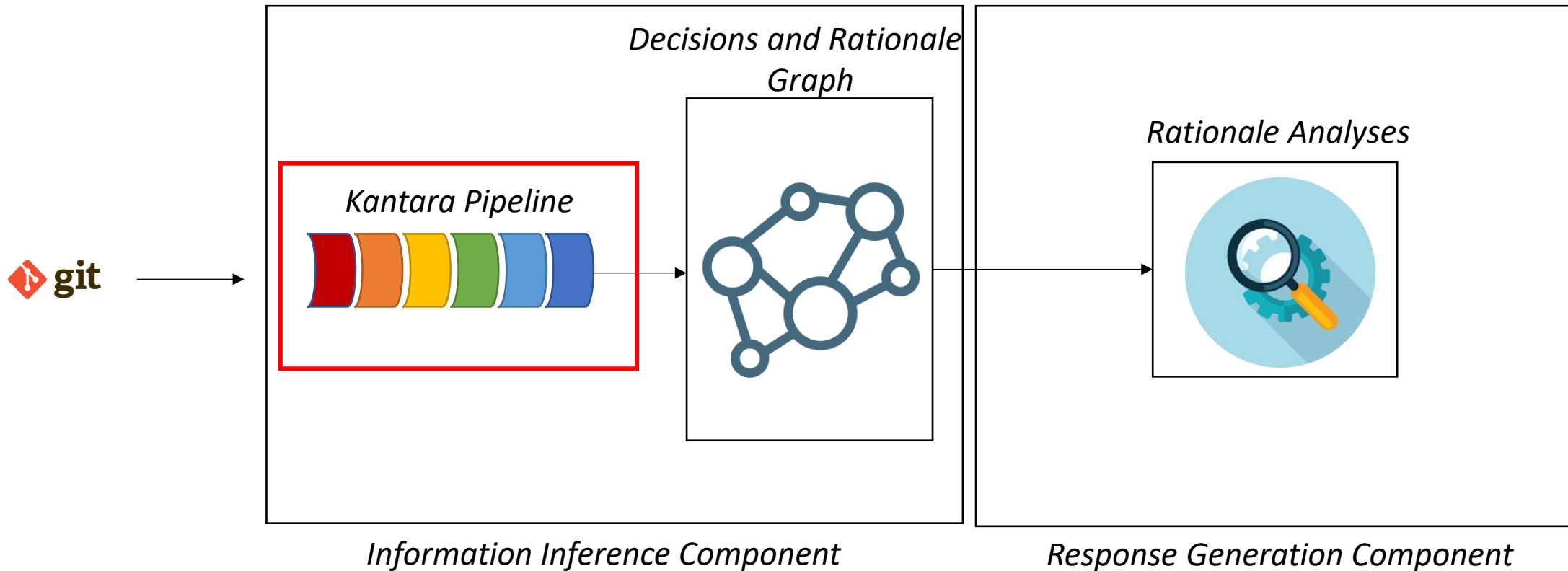
`remove is_linear_pte() to simplify code`

Decision

Rationale

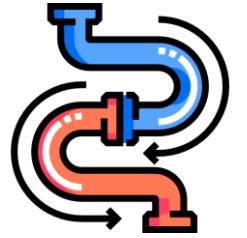
Rationale extraction and management system

- Based on the *On-Demand Developer Documentation (OD3)* concept [Robillard et al., 2017.]



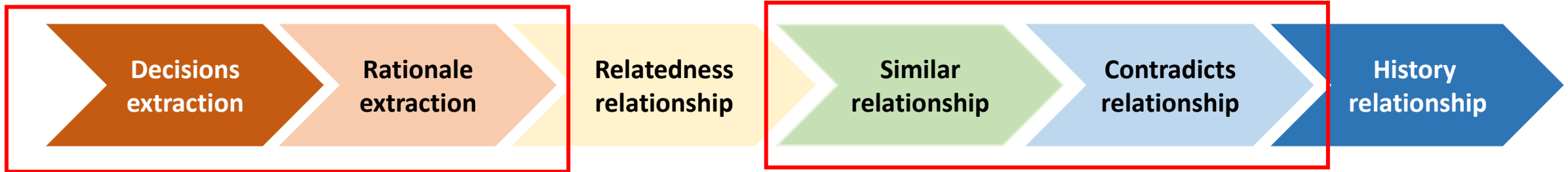
Kantara pipeline

Kantara **only** works on sentences containing **both** Decision and Rationale!



Extracting Decision-Rationale triples

Extracting Decision-Decision Relationships



to simplify code

rationale

`remove is_linear_pte()`

remove is_linear_pte() to simplify code

Rationale #1

rationale

Decision #1

Rationale #2

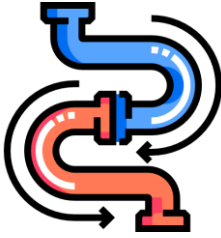
rationale

Decision #2

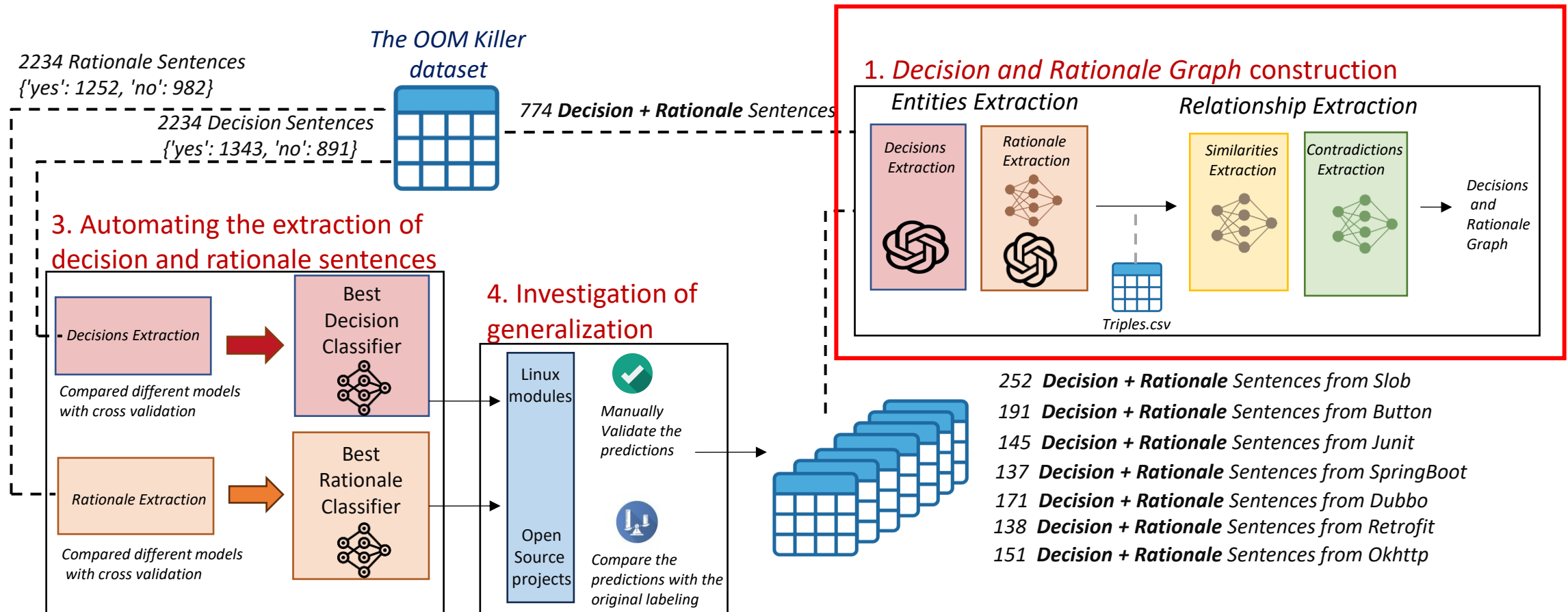
similar

contradicts

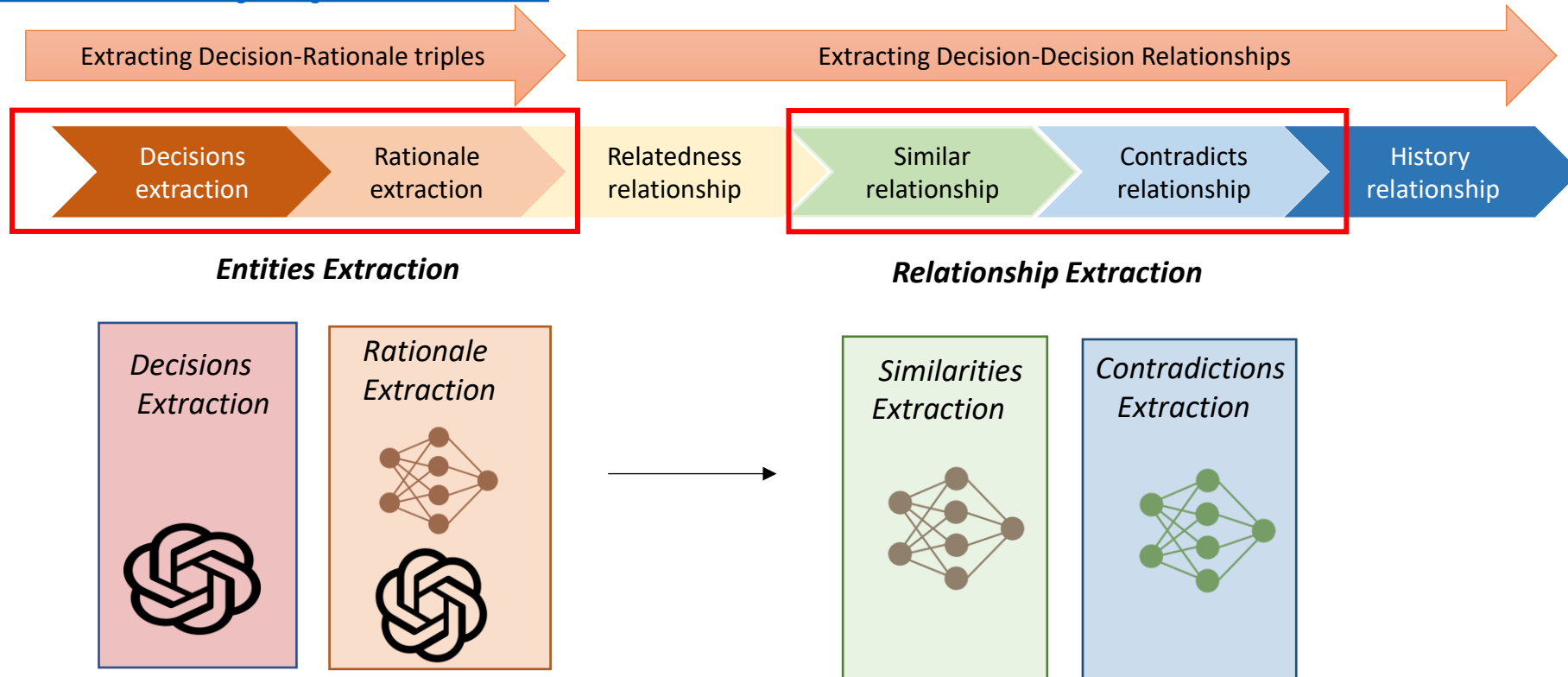
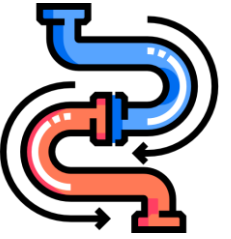
Kantara pipeline



Main contributions: Concrete implementation of Kantara + Investigation of its generalization.



Kantara pipeline



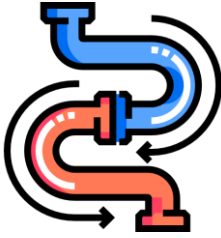
Entities Extraction: LLMs (ChatGPT)

Relationship extraction: Pre-trained models:

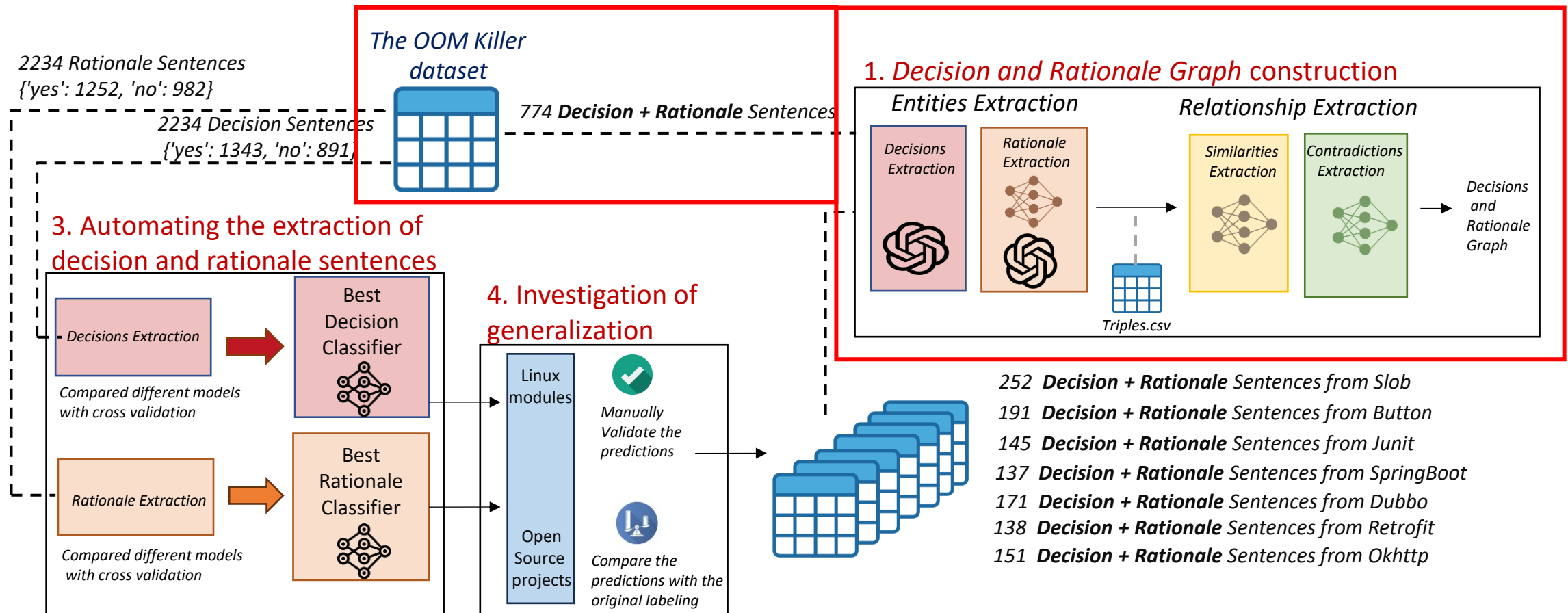
For **similarity** detection, *distilbert-base-nli-mean-tokens* model from the *sentence_transformers* library

For the **contradiction** detection, *roberta-large-mnli* model from the *transformers* library

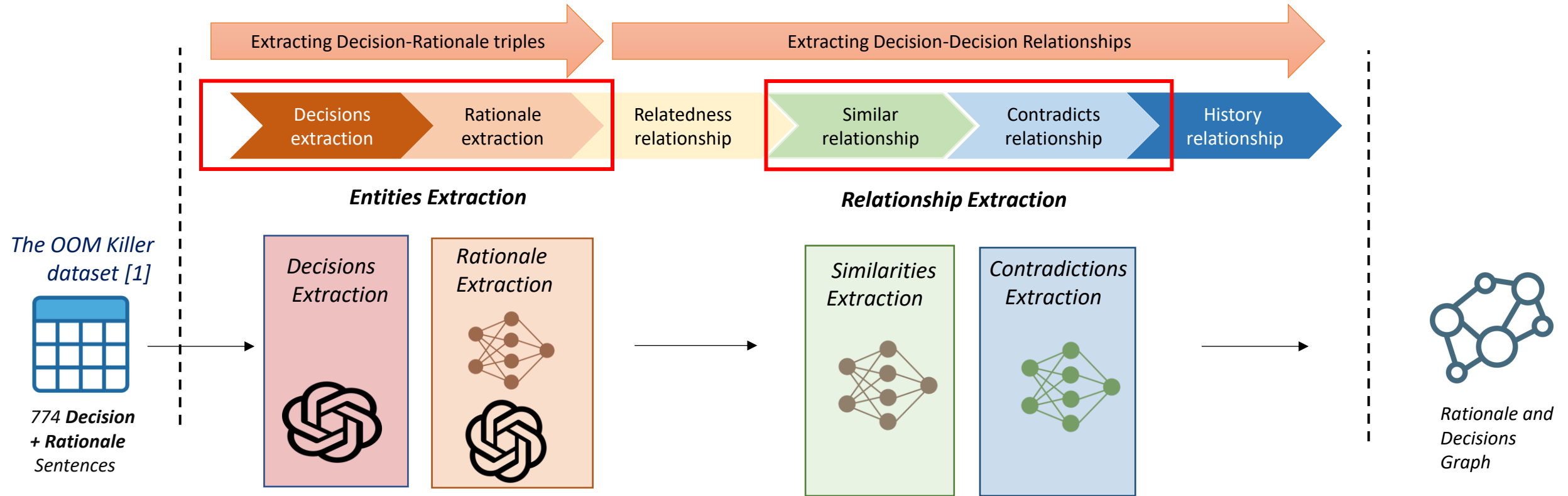
Kantara pipeline



Main contributions: Concrete implementation of Kantara + Investigation of its generalization.



Kantara pipeline

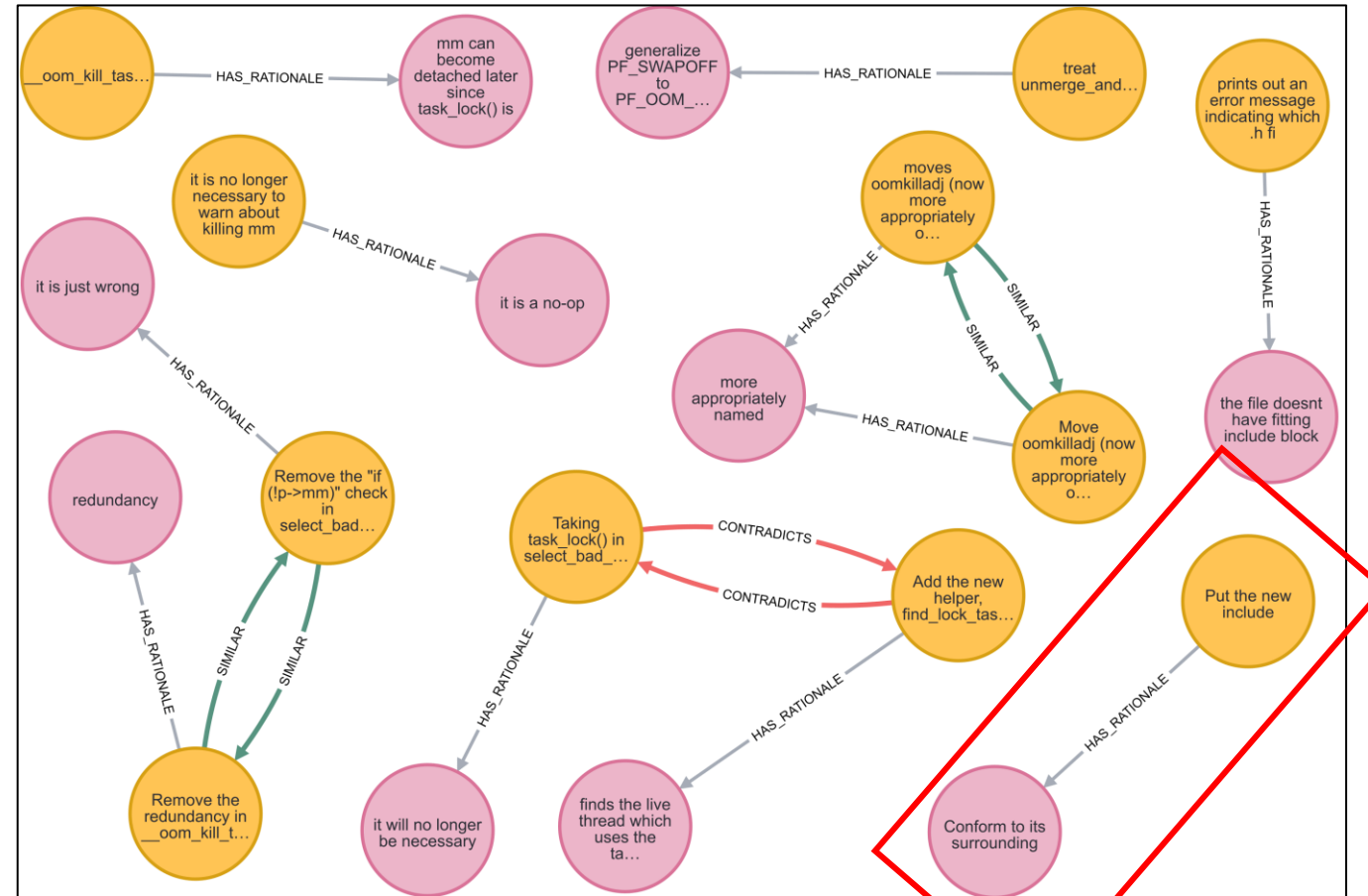


Kantara **only** works on sentences containing **both** Decision and Rationale!

Kantara output – OOM-Killer

- 774 **Decision + Rationale** Sentences
- 527 extracted decisions:
⇒ 138601 **pairs** of decisions

⇒ Computed these relationships between **each pair** of the extracted **decisions** with threshold of **0.9**
- 29 **similarities**
- 396 **contradictions**
- Performance: ~ 3 hours.



Part of the *Rationale and Decisions* graph for the OOM-Killer module.

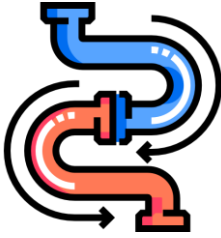
How to Generalize?



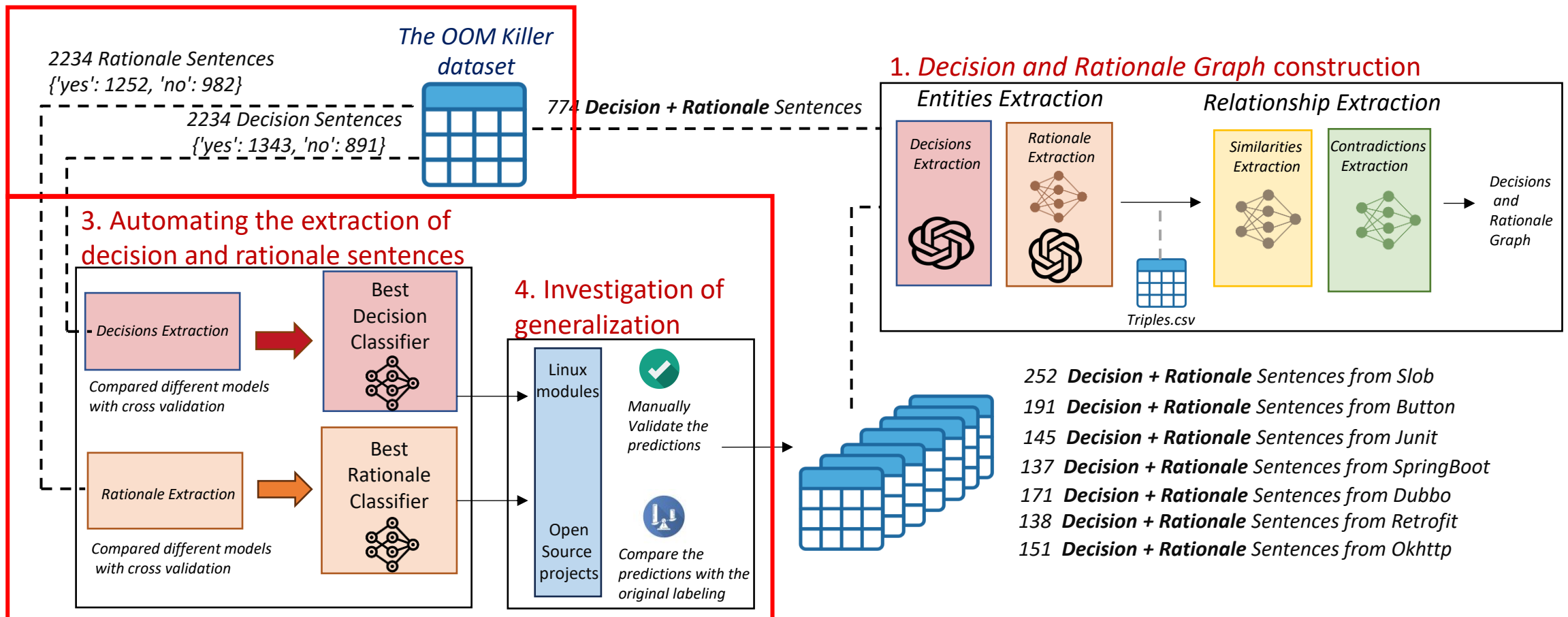
- Kantara **only** works on sentences containing **both** Decision and Rationale!
- We need sentences that are **both** decision and rationale
 - => We need classifiers that can generalize in other contexts

remove `is_linear_pte()` to simplify code

Kantara pipeline



Main contributions: Concrete implementation of Kantara + Investigation of its generalization.



Extractors



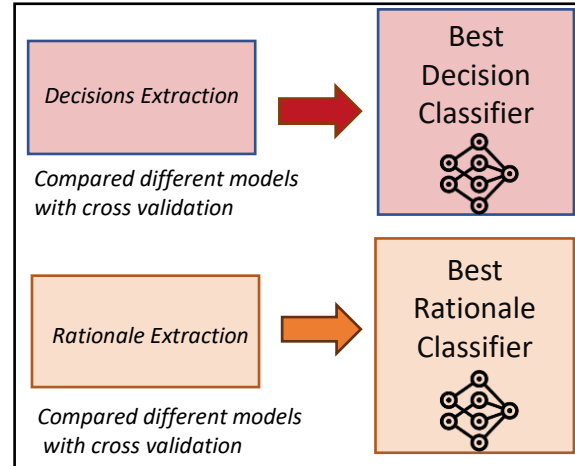
The OOM Killer dataset



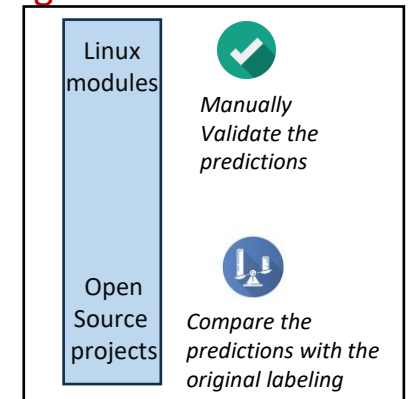
2234 **Decision** Sentences
{'yes': 1343, 'no': 891}

2234 **Rationale** Sentences
{'yes': 1252, 'no': 982}

3. Automating the extraction of decision and rationale sentences



4. Investigation of generalization



Two binary classification tasks:

- **Decision-containing** sentence classification
- **Rationale-containing** sentence classification

Several Binary classifiers:

- Trained on the OOM-Killer dataset
- 10-fold cross validation on the OOM-Killer dataset(evaluation) => Choose **Best** Models

Extractors generalizable?



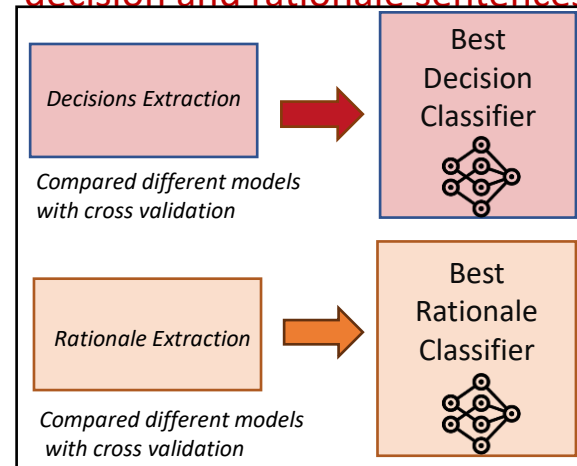
The OOM Killer dataset



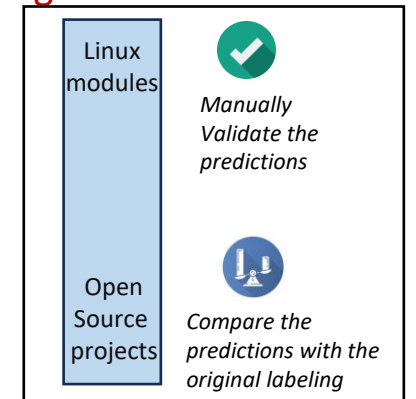
2234 **Decision** Sentences
{'yes': 1343, 'no': 891}

2234 **Rationale** Sentences
{'yes': 1252, 'no': 982}

3. Automating the extraction of decision and rationale sentences



4. Investigation of generalization

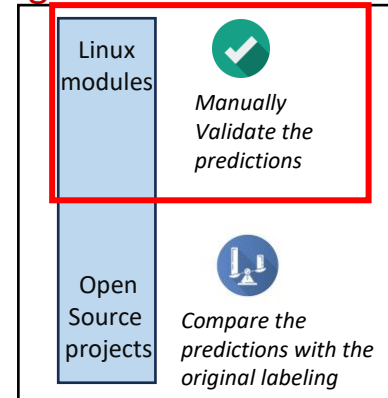


- 2 Linux modules: *slob.c* and *button.c*
- 5 Github projects: *Junit*, *Retrofit*, *SpringBoot*, *Dubbo*, *Okhttp*

Extractors generalizable? - Validation

Linux: *slob.c* and *button.c*

- post-label commit-based sampling
- random 20 commits per file / 91 sentences per file
- individually indicated whether or not we agree with the predictions of the classifiers



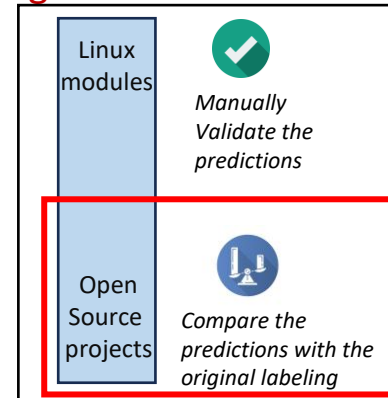
Rater	mm/slob.c		drivers/acpi/button.c	
	Decision	Rationale	Decision	Rationale
Rater 1	70.3%	69.2%	73.6%	80.2%
Rater 2	73.6%	70.3%	71.4%	80.2%
Rater 3	73.6%	80.2%	65.9%	73.6%
Average	72.5%	73.6%	70.3%	78%
Inter-rater agreement	79.1%	65.9%	75.8%	60.4%

The raters had a total agreement for about 60-79% sentences => The classifier is about as accurate as a human annotator.

Extractors generalizable? - Validation

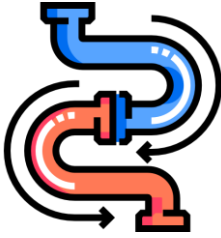
Other OS projects:

- Dataset of [Tian et al. *What Makes a Good Commit Message?*, ICSE'22]
- 1649 commit messages from five OSS projects.
- Labels: *Why_and_What*, *No_What*, *No_Why*, and *Neither*.
- Align labels to compare [**Why = Rationale** and **What = Decision**].

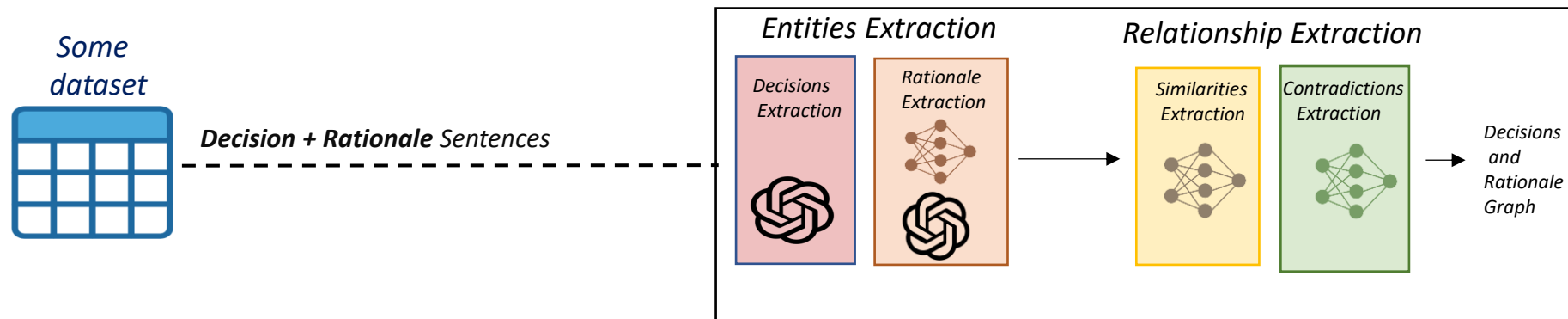


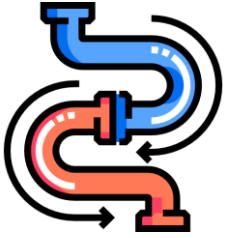
Project	Decision				Rationale			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
JUnit4	0.60	0.81	0.65	0.72	0.57	0.58	0.75	0.66
Apache Dubbo	0.57	0.72	0.70	0.71	0.66	0.77	0.78	0.78
Square Retrofit	0.64	0.92	0.66	0.77	0.57	0.55	0.80	0.65
Square OkHttp	0.57	0.87	0.59	0.71	0.62	0.71	0.77	0.74
Spring-boot	0.46	0.83	0.47	0.60	0.74	0.96	0.75	0.84
All commits	0.56	0.83	0.60	0.70	0.64	0.73	0.77	0.75

Kantara can generalize

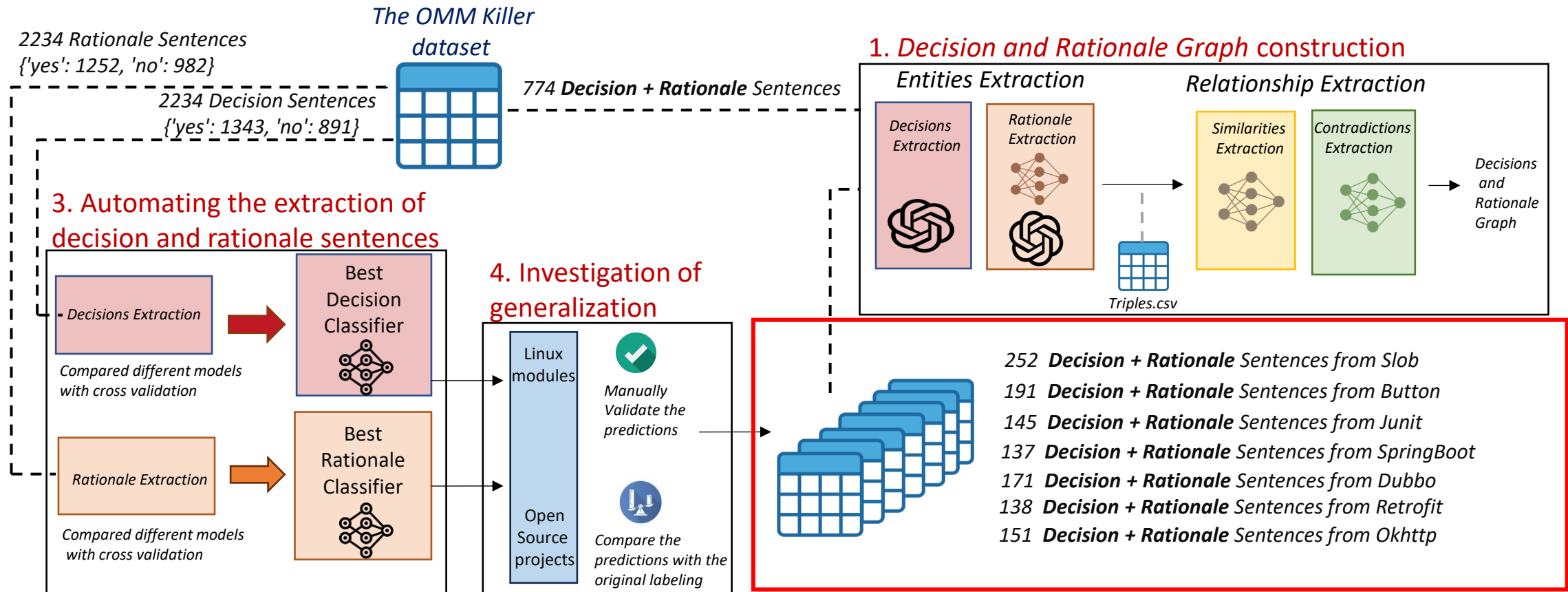


LLMs and pre-trained models + Reusable classifiers = Kantara can **generalize**





Application to Other Contexts



Semantic Rationale Analyses

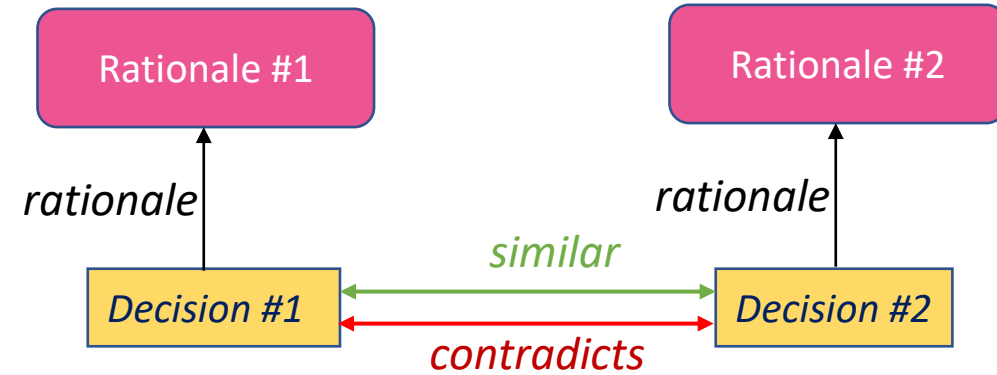
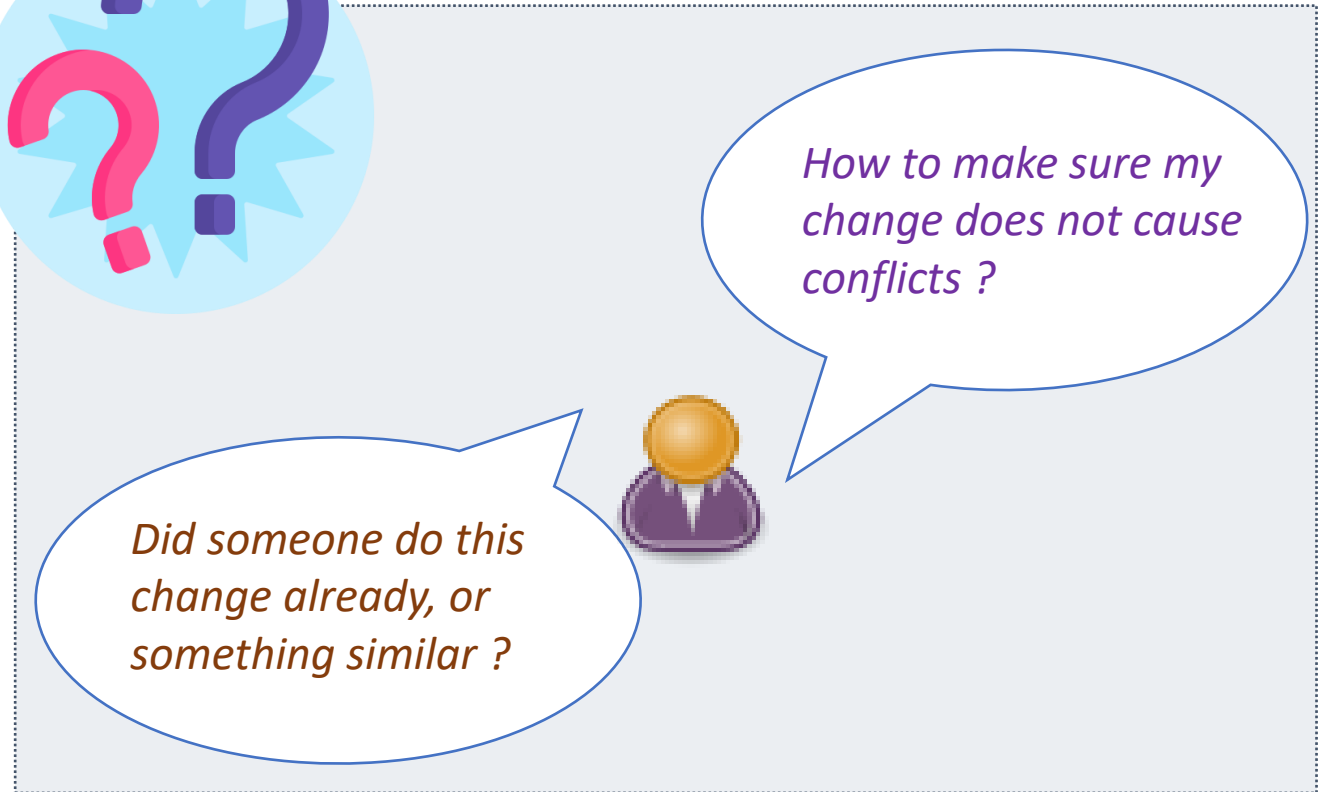
Decision-Decision Relationships.



- Two **similar** Decisions
 - Reuse past solutions



- Two **contradictory** Decisions
 - Avoid potential conflicts and reasoning problems



Semantic Rationale Analyses



- Reuse past solutions
 - Two similar Decisions

Nº	Decision 1	Decision 2	Relationship
1	Prepare for new header dependencies before moving code to <linux/sched/coredump.h>	Prepare for new header dependencies before moving code to <linux/sched/mm.h>	Similar (0.99)
2	Create a trivial placeholder <linux/sched/coredump.h> file that just maps to <linux/sched.h> to make this patch obviously correct and bisectable	Create a trivial placeholder <linux/sched/mm.h> file to make this patch obviously correct and bisectable	Similar (0.94)
3	Remove unnecessary locking in exit_oom_victim()	remove unnecessary locking in oom_enable()	Similar (0.93)
4	moves oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct signal_struct	Move oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct mm_struct	Similar (0.97)

Semantic Rationale Analyses



- Avoid potential conflicts
 - Two **contradictory** Decisions

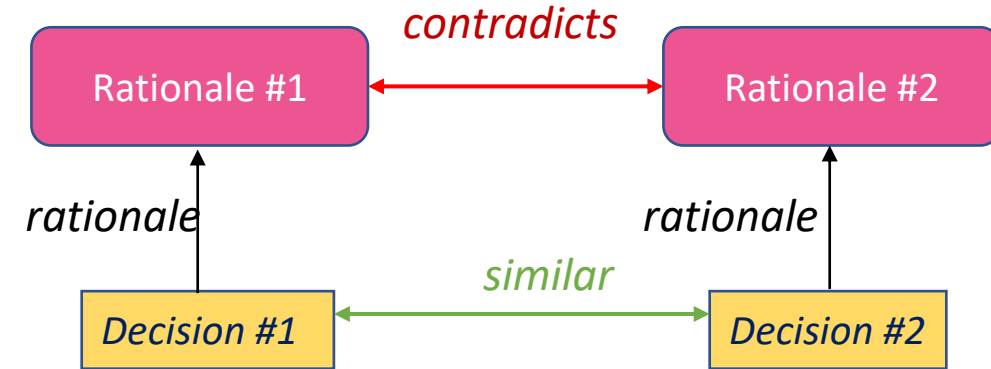
Nº	Decision 1	Decision 2	Relationship
5	replace try_oom_reaper by wake_oom_reaper	This patch adds try_oom_reaper.	Contradicts (0.98)
6	push the re-check loop out of freeze_processes into check_frozen_processes	We are not checking whether the task is frozen	Contradicts (0.96)
7	giving the dying task an even higher priority	to avoid boosting the dying task priority in case of mem_cgroup OOM	Contradicts (0.97)
8	Add the new helper, find_lock_task_mm()	Taking task_lock() in select_bad_process() to check for OOM_DISABLE and in oom_kill_task() to check for threads sharing the same memory will be removed in the next patch in this series	Contradicts (0.91)
9	Move oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct mm_struct	make the magical value of -17 in /proc/<pid>/oom_adj defeat the oom-killer altogether	Contradicts (0.95)

Semantic Rationale Analyses

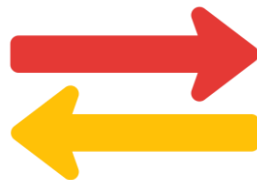


Inconsistency Detection Mechanisms.

- Two **similar** Decisions & **contradictory** Rationales
- Two **contradictory** Decisions & **similar** Rationales



- Validate the integrity of the graph and clean up nonsensical relations
- Reveal miscommunications and inaccurate documentation
 - Insufficient understanding of the impact of the proposed decision
 - Wrong description of a patch in a commit message
 - Malicious attempts to hide suspicious code contributions



Semantic Rationale Analyses

Two **contradictory** Decisions & **similar** Rationales

Project	Decision 1	Rationale 1	Decision 2	Rationale 2	D/D Relation	R/R Relation
Slob.c	move it out of the slab_mutex	which we have to hold for iterating over the slab cache list	slab_mutex for kmem_cache _shrink is removed	after its applied, there is no need in taking the slab_mutex	Similar (0.806)	Contradicts (0.900)
Button.c	Setting lid_init_state to ACPI_BUTTON _LID_INIT _OPEN on the T200TA	fixes the unwanted behavior, adds a DMI based quirk	libinput (1.7.0+) can fix the state of the lid switch for us: you need to set the udev property [...] to write_open	can fix the state of the lid switch for us	Contradicts (0.860)	Similar (0.687)
Dubbo	fix typo:<iden> (<pr_link>)	typo	Support <iden> auto recognize in <file_name> (<pr_link>)	Enhancement Decision: Fix <issue_link>	Similar (0.8110)	Contradicts (0.627)
Retrofit	Propagate call-back executor even when not explicitly specified	This lets custom <iden> implementations use the <file_name> default for their callbacks	Switch to a generic to/from <file_name> , explicit factory	This allows for special-ization based on for what purpose the <file_name> is being created.	Contradicts (0.891)	Similar (0.776)

Future Work

- Integration with development platforms
- Improve the approach (more advanced LLMs, include rationale spanning multiple sentences)
- Enrich the graph (more relationships) => more semantics
- Integrate other data sources (discussions, code reviews, etc..)
- Application on more projects => Repository of Rationale Information.

Automated Extraction and Analysis of Developer's Rationale in Open Source Software

