

Towards Understanding and Analyzing Rationale in Commit Messages using a Knowledge Graph*

Context

- Extracting rationale from commit messages is useful [1].
- But it is difficult (rationale is embedded in natural text) [2].
- Ongoing work on the Kantara pipeline for end-to-end rationale reconstruction to **extract, structure and analyze** rationale [3].
- **Main contribution: Showcasing the practicality of Kantara for a real-world software project.**

Dataset Creation

- A labelled dataset of 180 commit messages for the Out-of-Memory component of the Linux kernel:
 - 3 annotators
 - Codebook (Table 1).
 - Fleiss Kappa ~ 0.69.
 - Sentence-based labelling
 - Multi-labels (Figure 1).

Label	Meaning
Decision	An action or a change that has been made, including a description of the patch behaviour
Rationale	Reason for a decision or value judgment
Supporting Facts	A narration of facts used to support a decision
Inapplicable	Pre-processing error or bad sentences (i.e., does not contain English sentences)

Table 1. Codebook.

Rationale Extraction

- Widely-used models for sentence classification
- **Binary classification**
 - Low recall
 - The Decision Tree algorithm gave the best results.
 - Decision classification better than Supporting Facts classification.
 - Low performance for Rationale classification
- **Multi-label classification**
 - XGBoost classifier [4] gave the best performance.
 - For XGBoost, the Decision classification was better than Supporting Facts classification. Rationale classification was the worst.

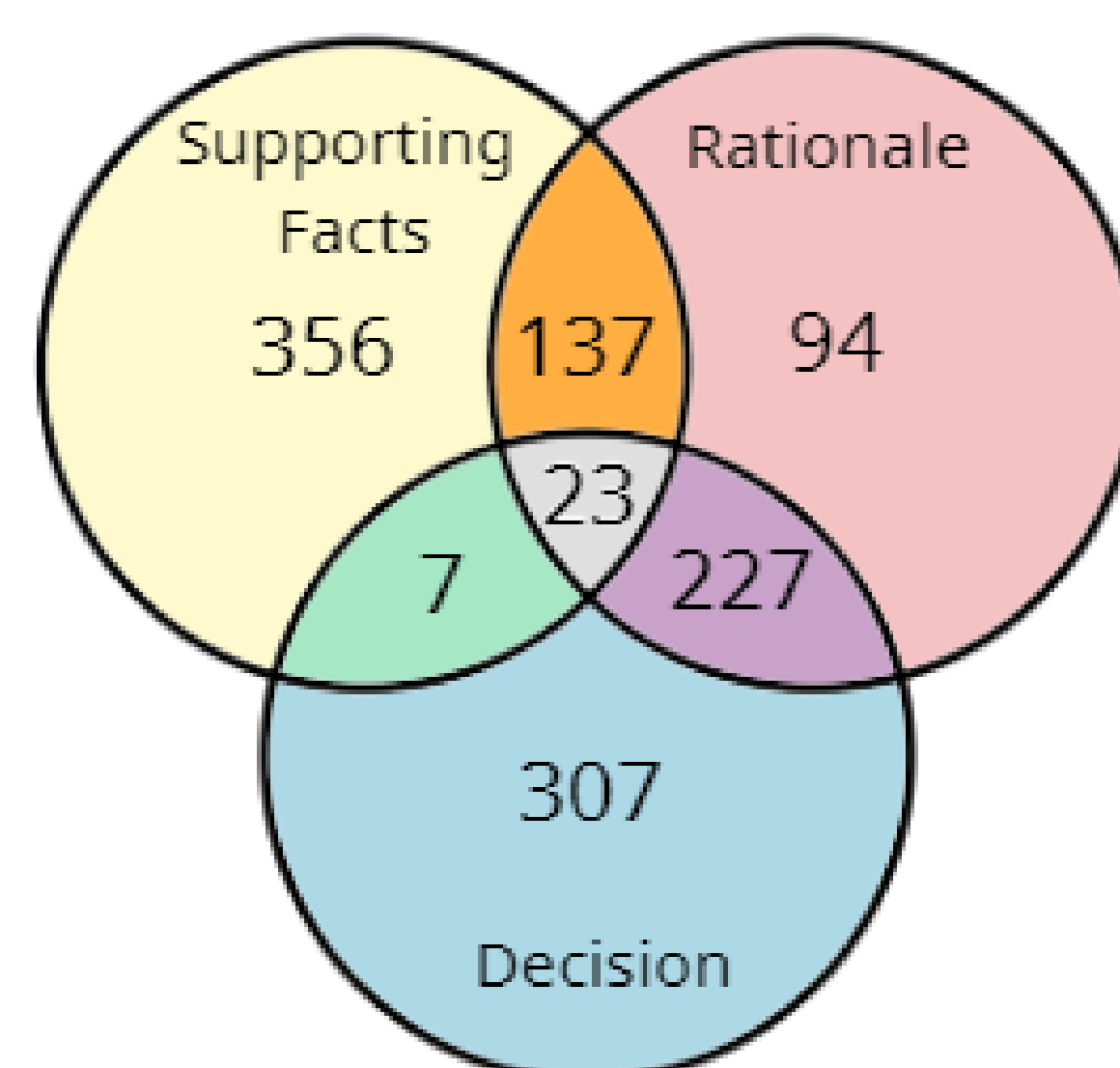


Figure 1. Distribution of the labelled sentences

Rationale Structuring

- An ontologically-based knowledge graph (Figure 2)
- For concrete implementation, the openCAESAR framework [5]

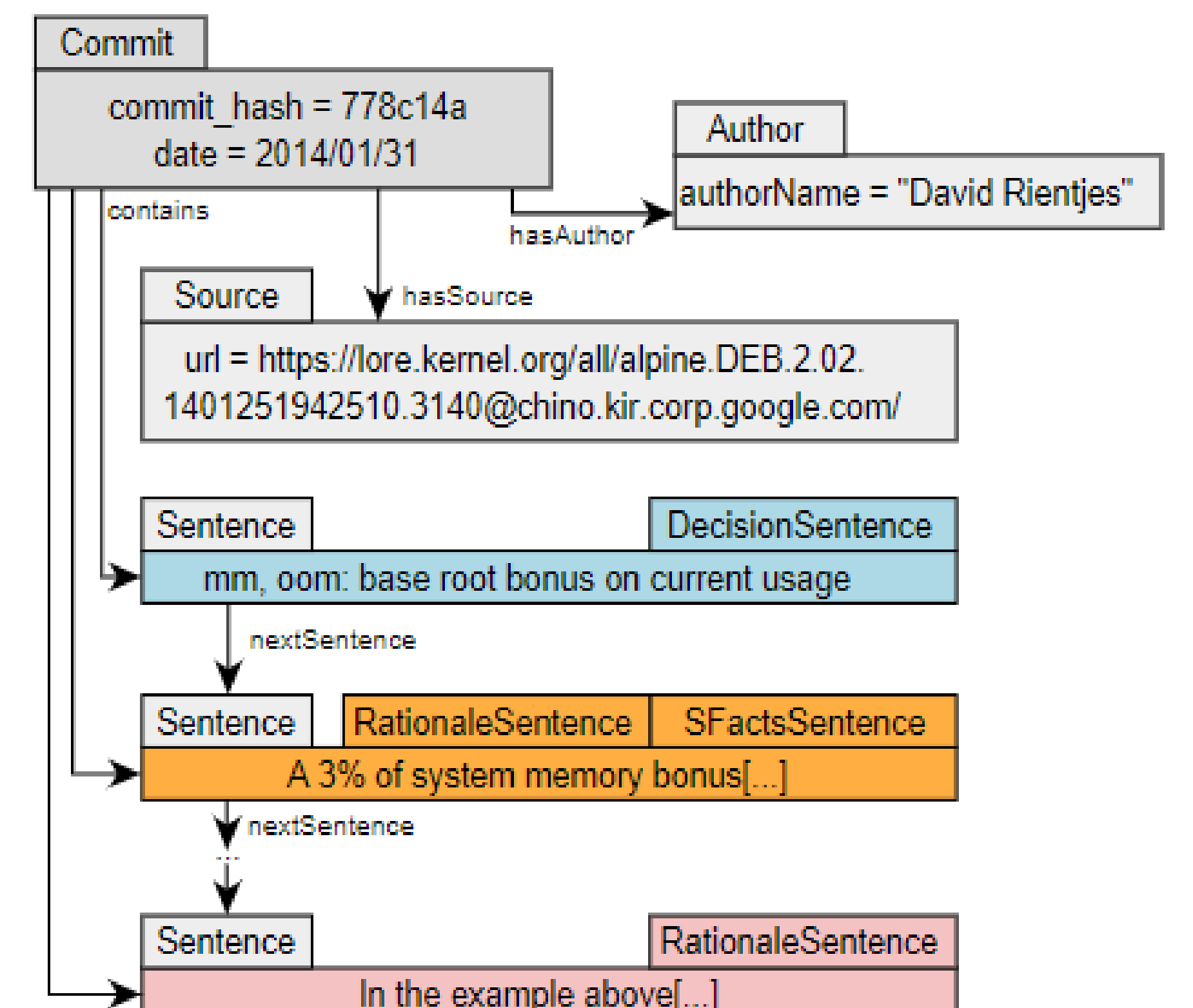


Figure 2. Decision/Rationale Graph

Rationale Analysis

Leverage ontological semantics for

- Additional reasoning
- Graph queries
- Interactive visualizations (Figure 3).

Challenges and Future Work

1. Expanding the Rationale Ontology
2. Subjective labeling
3. Classification performance
4. Generality

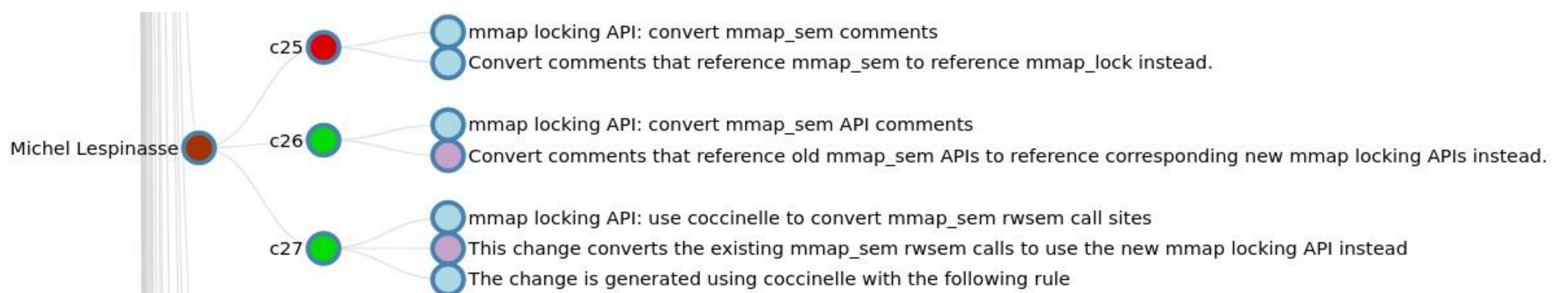


Figure 3. Interactive visualization of authors, their commits, and commit sentence text

*M. Dhaouadi, B. J. Oakes, and M. Famelis, "Towards Understanding and Analyzing Rationale in Commit Messages using a Knowledge Graph," in the ACM/IEEE 26th International Conference on Model-Driven Engineering Languages and Systems: Companion Proceedings, 2023.

Contact

Mouna Dhaouadi
 Université de Montréal
 Email: mouna.dhaouadi@umontreal.ca
 Website: <https://www-labs.iro.umontreal.ca/~dhaouadm/>

References

1. J. E. Burge, J. M. Carroll, R. McCall, and I. Mistrik, Rationale-based software engineering. Springer, 2008.
2. Y. Liu, Y. Liang, C. K. Kwong, and W. B. Lee, "A new design rationale representation model for rationale mining," Journal of Computing and Information Science in Engineering, vol. 10, no. 3, 2010.
3. M. Dhaouadi, B. J. Oakes, and M. Famelis, "End-to-end rationale reconstruction," in 37th IEEE/ACM International Conference on Automated Software Engineering, 2022, pp. 1–5.
4. T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou et al., "XGBoost: extreme gradient boosting," R package version 0.4-2, vol. 1, no. 4, pp. 1–4, 2015.
5. M. Elaasar, N. Rouquette, D. Wagner, B. J. Oakes, A. Hamou-Lhadj, and M. Hamdaqa, "openCAESAR: Balancing agility and rigor in model-based systems engineering," in Proceedings of the System Analysis and Modelling (SAM) Conference, in the 26th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, 2023, accepted.