

MatSci YAMZ: Integrating Al into Metadata Dictionaries





Institute for Data
Driven
Dynamical
Design
ID

Addy Ireland¹, Scott McClellan², John Kunze², Christopher Rauch², Joel Pepper², Colton Gerber³, Mat Kelly², and Jane Greenberg²

¹Pennsylvania State University, State College, PA, ²Metadata Research Center, College of Computing & Informatics, (CCI), Drexel University, Philadelphia, PA

³Toberer Research Group, Colorado School of Mines, Golden, CO

YAMZ

- Collaborative vocabulary application
- Allows users to define terms related to their field
- Community can comment and vote on terms similar to Stackoverflow

Motivation

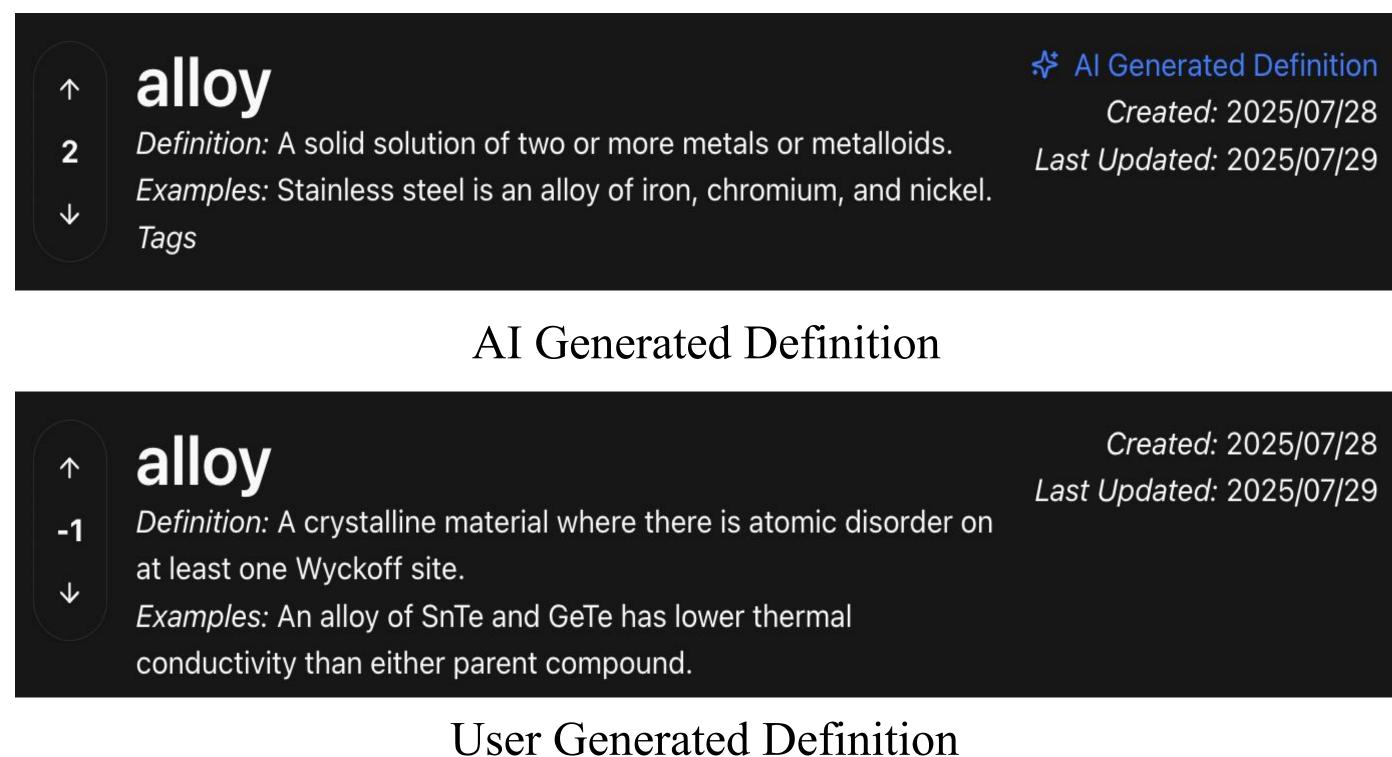
- Improve vocabulary usage among material scientists by reducing ambiguity in meaning
- Leverage artificial intelligence (AI) to assist in creating and improving definitions

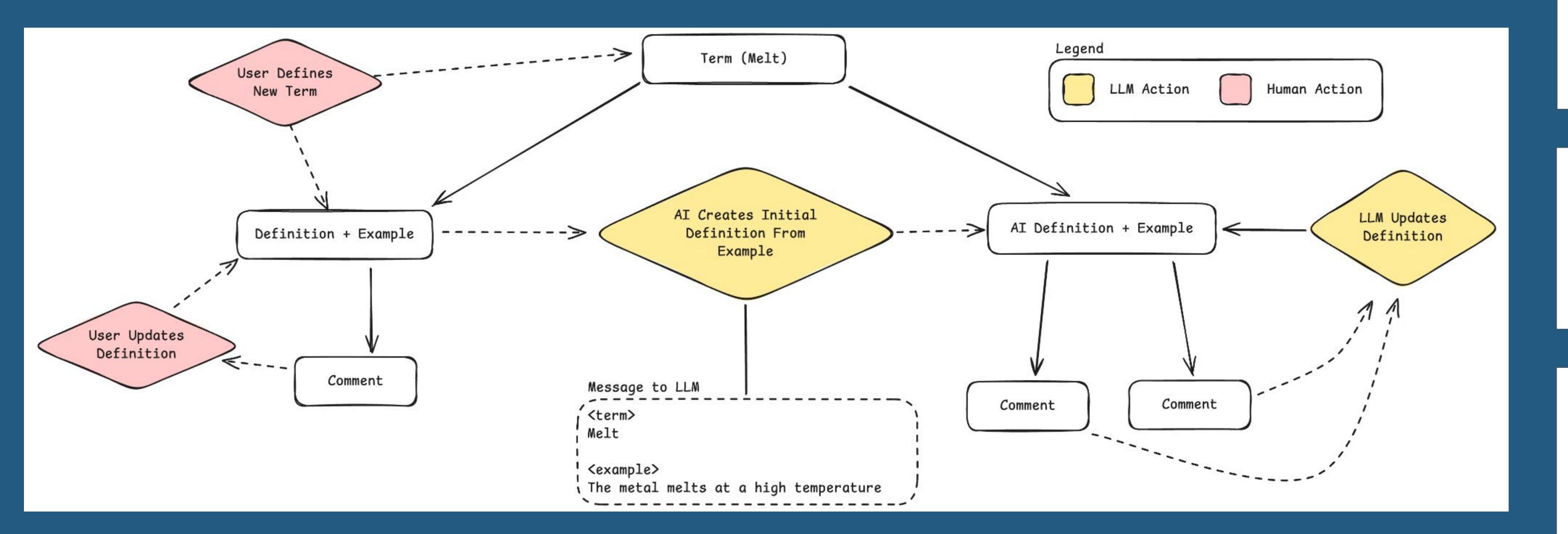
How MatSci YAMZ Works

- Users can create and discuss definitions related to material science.
- Integrates large language model (LLM) Al into the definition creation workflow
- Al generates definitions based upon the users' examples
- Al definitions are automatically generated any time a new term is defined
- Al will revise its definition whenever it receives a new comment

Implementation

- Written in TypeScript using NextJS with tRPC for communication between server and client. Using PostgreSQL to store data.
- Ollama used to run and interact with LLM's through the Ollama API, making it incredibly easy to switch models (see table to right)
- When a new term is defined, we feed the term and example into the LLM, asking it to create a definition. The output is then posted on MatSci YAMZ as a definition
- When a user comments on a AI definition, the comment is added to the LLM chat history for the definition and the model is re-run. The AI definition on MatSci YAMZ is then updated to reflect the changes.





Considerations

- We used open source models to avoid costs associated with running proprietary models
- We picked Gemma3 based on speed and accuracy. We benchmarked 3 of the top open-sourced Al Models: DeepSeek-R1, Gemma3, Qwen3
- We used Ollama to quickly download, and run the large language models, which also gave us flexibility to separate the LLM from the web server
- To prompt the model, we used a combination of structured outputs and a system prompt describing what the models role is

Model	Output Tokens	Tokens Per Second	Total Duration
gemma3:27b	40	5.24	8.56s
deepseek-r1	52	1.87	30.84s
qwen3:32b	46	3.79	14.37s

Future plans

- Testing with material scientists
- Broader integration of AI into workflow
- Hot swappable models
- Improving the user interface and experience
- Building out admin page to manage users, definitions, and terms

Conclusions

- Broader testing with more material scientists
- Broader integration of Al
- Using different LLM models

Give it a try!

- Go to <u>matsci.yamz.net</u> or scan --->
- Login with Google
- Define a term or give feedback on a existing definition



Acknowledgments

NSF-HDR-OAC #2118201 Institute for Data Driven Dynamical Design, and Robert Sammarco, Drexel COE, and Brian Bijeau, CCI/Drexel

Sources

- Gemma 3: https://arxiv.org/abs/2503.19786
- Deepseek R1: https://arxiv.org/abs/2501.12948
- Qwen3-32B: https://arxiv.org/abs/2505.09388