Tech Credit Final Deliverable

Pusen, Mallory, Jules, Amelia, Chloe

A few reminders...

Our Research Question

Is it possible to identify different types of technical credit using an LLM?

What is Technical Credit?

Tech credit (TC) is the recognition of positive, strategic design decisions that make a system easier to maintain, scale, or improve over time. It shifts focus from just avoiding technical debt to valuing investments that strengthen a system's long-term quality and adaptability.

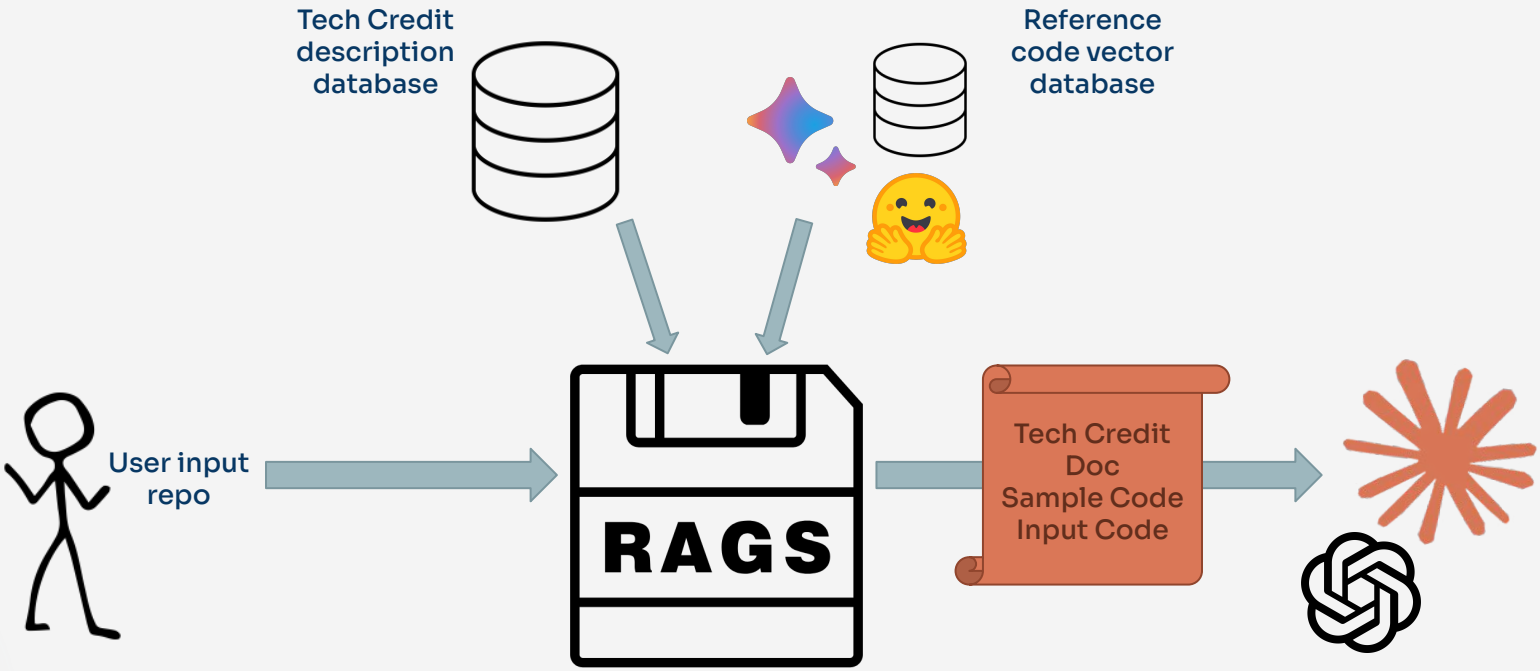
Use Cases

Technical Credit ensures software can evolve smoothly as both **business needs** and **technologies shift**

Faster development, easier change, and more resilient systems

Real-World Examples Made Easier by TC:
Switching cloud providers, integrating new gateways, onboarding new developers, launching platforms from web to mobile, adding new business logic quickly

Our System Architecture



Why Is It Important

Technical credit is important because it encourages and rewards forward-thinking decisions that improve a system's long-term *sustainability, quality, and adaptability*.

Identifying TC in software can allow you to make a better evaluation of a system's *architectural foresight*.

Project Aims

Build an **explainable prototype system** that identifies Technical Credit artifacts with **clear methodology** and a foundation for **future work.**

Overall Pipeline Architecture

🔑 Auth Setup
Load API keys for LangSmith, Anthropic, Google, GitHub, HuggingFace

💬 LLM & Embeddings

Init Claude 3.5 Haiku model OpenAI 4.1 mini via LangChain

Use Google or HuggingFace Embeddings for vectorization

📁 Data Sources

Pull Python files via GithubFileLoader

Load custom JSON metadata

🧠 Vector Store

Store and embed code/docs using Chroma

🧩 Metadata Mapping

Map file paths to annotated tech credit descriptions

Training/Embedding

1. Repo & Doc Loading
⭐ Use GithubFileLoader to fetch code
⭐ Load JSON metadata for annotations

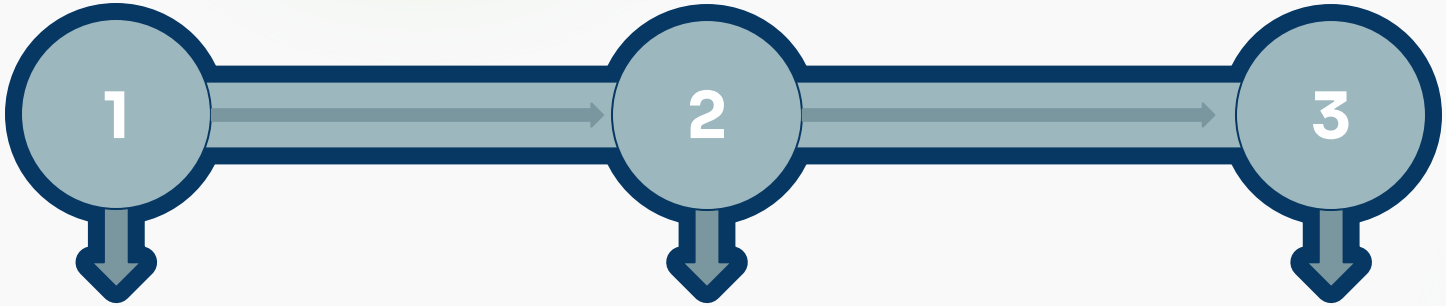
2. Document Splitting
⭐ Use TiktokenSplitter to chunk code/doc content
⭐ Prepare documents for embedding

3. Embedding Generation
⭐ Use GoogleGenerativeAIEmbeddings (text-embedding-004) or HuggingFaceEmbeddings (sentence-transformers/all-mpnet-base-v2)
⭐ Apply to both source code and document content

4. Vector Store Population
⭐ Store embedded chunks in Chroma DB
⭐ Two collections: code and document

5. Retrieval + Scoring
⭐ Use similarity search (e.g., cosine)
⭐ Retrieve top-k relevant documents per query

Prompt Design



Insert tech credit description documents

List of 3 most possible tech credit code snippets with example code from the training repo, and metadata of tech credit

Insert questions from users

Demo!

Deliverable!

Results

It is possible to have an LLM identify Technical Credit

-> When trained with appropriate examples and definition LLM Models can correctly identify TC
-> This is possible on small/simple python github projects

Not all LLMs are made the same

-> Claude and Google Gemini had the best performance
-> HuggingFace = 👎

There is room to improve this POC

-> The italians still have something to work on
-> more training data, bigger repos, exact heuristics for similarity

Limitations

- Worked with a small, domain-specific dataset, leading to biases toward generalized patterns.

- False positives risk inflating the frequency of detected Technical Credit, affecting result validity.

- Current system relies on a single vector database and monolithic classification approach.

Future Work

Expand datasets across more domains and project types to improve generalizability.

Enhance testing and validation with empirical metrics

Integrate MoE into the RAG pipeline: divide specialized services to handle different styles of software artifacts —> more targeted, “stage-specific” recommendations

Evolve into a “TC Advisor”, a system that not only detects but *prescribes* project-specific improvements

Learnings

Understanding Tech Credit

- Learned how to identify and articulate positive design decisions that add long-term value to a system.

Shifting Perspective

- Gained insight into framing software quality not just in terms of avoiding debt, but also recognizing credit.

What Was Hard

- It was challenging to condense a nuanced idea into a concise, consistent definition.

What Was Easy

- Finding examples of technical credit and understanding its value came naturally once we had the framework.

Thanks!

Reference: Gorton, Bucaioni, Pelliccione. "Technical Credit," Communications of the ACM, 2025.