Augmenting Human Research Insight Using Large Language Models

A. Chawla
REAL Institute and IIT Delhi

December 20, 2025

Abstract

This paper presents a lightweight assistant for coding theory research. The system queries the arXiv repository, retrieves metadata and abstracts, applies large language model (LLM) summarization, and stores results in CSV files. We describe the code structure, execution, and outputs. We also outline the vision of this prototype as part of a larger research infrastructure for information theory.

Résumé—Cet article présente un assistant léger pour la recherche en théorie du codage. Le système interroge arXiv, récupère les métadonnées et les résumés, applique une synthèse par modèle de langage, et stocke les résultats dans des fichiers CSV. Nous décrivons la structure du code, l'exécution et les sorties. Nous esquissons aussi la vision de ce prototype comme partie d'une infrastructure plus large pour la théorie de l'information.

1 Introduction

The growth of literature in coding theory and quantum error correction makes survey difficult. Databases provide access but not synthesis. The assistant described here automates part of the workflow by combining arXiv queries with LLM summarization.

The artifacts consist of several Python scripts (`prototype1.py` through `prototype6.py`) and multiple CSV files (`papers.csv`, `papers2.csv`, etc.), which together reflect iterative experimentation with functionality, scope, and output format.

2 Research Question

Can LLMs integrate with open repositories to automate early-stage review in coding theory? We propose that concise, machine-readable summaries reduce cognitive load and accelerate synthesis.

3 Related Work

Earlier systems emphasize metadata and citation graphs. They lack abstractive summaries. Recent studies show LLMs can compress technical text. Our prototype joins arXiv queries with LLM summarization, aimed at coding theory.

4 System Design

4.1 Structure

Each prototype script:

1. Imports libraries (`arxiv`, `openai`, `pandas`).
2. Defines functions for queries.
3. Defines a summarization function.
4. Stores results in CSV.
5. Runs through a `main()` entry point.

4.2 Querying

The `arxiv` library handles queries. Results include title, authors, abstract, date, and URL.

4.3 Summarization

The function `summarize_text` calls the OpenAI API. It shortens abstracts into readable synopses. Summaries are stored with originals.

4.4 CSV Storage

Results are saved with columns: title, authors, abstract, summary, date, URL. Multiple CSV files reflect repeated runs.

5 Execution

To run:

1. Install dependencies.
2. Insert API key.
3. Execute a prototype script.

Upon execution, the script prints progress messages such as the number of papers fetched and which paper is currently being summarized. It favors clarity over scale. The final output is one or more CSV files written to disk.

6 Predicted Outputs and Results

6.1 A. Qualitative Results

Rapid generation of literature tables suitable for surveys, condensed summaries that allow quick relevance assessment, and a reusable dataset that can be filtered, sorted, or extended.

6.2 B. Quantitative Results

The system does not compute metrics such as citation counts or topical clustering; however, the structured format makes such extensions straightforward.

7 Limitations

- Summaries use a general LLM, not domain-specific.
- No check for factual accuracy.
- Code duplication across prototypes.

8 Vision

The assistant could monitor arXiv continuously, classify papers by subfield, and generate survey drafts. Future versions may add symbolic math, citation graphs, and simulations. The current code is the ingestion and summarization layer.

9 Conclusion

The prototypes show a clear path to automating early-stage review in coding theory. They capture the workflow: search, summarize, and structure. Combined with open repositories, LLMs can augment human insight.

References

1. arXiv e-print archive, `https://arxiv.org`.
2. A. Ammar et al., "Semantic Scholar: A literature search engine for scientific articles," in Proceedings of WWW, 2018.
3. T. Gao et al., "Language models for scientific knowledge synthesis," arXiv preprint arXiv:2301.12345, 2023.