**Enhancing AI Explainability: Summarization Assistant for Academic Research**

**Abstract:**

This project aims to build an intelligent summarization system that processes full-text academic papers and generates concise, user-tailored summaries. Leveraging pre-trained transformer models such as T5, BART, or BERT, the system will be fine-tuned specifically on scientific literature sourced from open-access journals and datasets like arXiv and PubMed.

The tool will enable users to generate summary for the whole article or individual sections of the paper (i.e., Introduction, Methods, Results, or Conclusion). Beyond the usual AI summarization tools, which often act as black-boxes, this project will inform the user that which specific portions of the original paper contributed most significantly to the generated summaries. By providing transparency and explainability into the summarization process, the system aims to enhance user trust and satisfaction, making it not just a summarizer but an interpretable assistant for academic research workflows.

Additional objectives include benchmarking the system's output quality against traditional LLM-based summarization tools (e.g., Claude, Elicit), using metrics such as ROUGE and BERTScore, and developing a lightweight, accessible interface through Streamlit or Gradio for end-users.

Pre req: Python programming, basic NLP concepts, familiarity with Hugging Face Transformers, and experience working with machine learning models

**Step 1: Define Scope and Use Case**

**Objective:**  
Summarize academic articles by section (Introduction, Methods, Results, etc.) or generate entire abstracts from full-texts.

**Choices to make:**

* Input source: PDFs, plain text, or structured data (e.g., JSON from arXiv)
* Output format: Section summaries or single abstract
* Domain: General science or medical (PubMed)

**Step 2: Dataset Selection and Preprocessing**

**Recommended Datasets:**

* **arXiv dataset** — scientific papers and abstracts
* **PubMed dataset** — biomedical research
* [**SciSummNet**](https://github.com/WING-NUS/scisumm-corpus) — scientific summarization benchmark

**Preprocessing:**

* Tokenization and section splitting
* Truncation (for model input size constraints)
* Create (document, abstract) pairs or (section, summary) pairs

**Step 3: Choose and Load Pretrained Model**

Options:

* BART (best general abstractive summarization model)
* T5 (flexible text-to-text)
* PEGASUS (specially pretrained for summarization)

Check Hugging Face

**Step 4: Fine-Tune the Model**

Use the Hugging Face Trainer API for easy training.

**Step 5: Evaluate Summary Quality**

**Metrics:**

* ROUGE (Recall-Oriented Understudy for Gisting Evaluation)
* BERTScore (semantic similarity with contextual embeddings)

**Step 6: Explainability**

Add explainability: why a sentence was chosen (use SHAP for NLP or attention maps)

**Step 7: Build Section-Wise Summary Interface**

Use Gradio or Streamlit (with explanibility)

**Step 8: Compare with Claude, ChatGPT, Elicit**

* Choose 3 sample papers
* Run them through your assistant and the others
* Compare summaries using ROUGE and human evaluation (fluency, factuality, readability)

# Pseudo-architecture

1. Summarization model: BART/SciBERT fine-tuned on academic papers

2. Explainability layer:

- Primary: Attention visualization

- Secondary: LRP for important/disputed claims

- Tertiary: Extractive alignment validation [BERT-based extraction]

3. UI: Interactive dashboard with source-summary linking [OPTIONAL]

Summary generates: "The authors claim their method is 10x faster"

Layer 1 (Attention): Shows attention on "efficiency", "speed", "performance"

→ Insight: Model is looking at performance-related terms

Layer 2 (LRP): Reveals main contribution from "10-fold reduction in time"

→ Insight: The "10x" specifically comes from one sentence, not averaged

Layer 3 (Extractive): Finds exact match "10x speedup over baseline"

→ Insight: Claim is directly supported, not hallucinated

[Summary Display]

"The study introduces a novel architecture achieving 95% accuracy [🔍][📊][✓]"

[🔍] Click for attention view - Shows which parts model focused on

[📊] Click for LRP analysis - Shows true contribution scores

[✓] Click for validation - Shows extractive support