"Explaining Relationships Among Research Papers" (2024) a **feature-based, prompting-oriented approach** using large language models (LLMs) to generate coherent, multi-paper citation paragraphs for literature reviews The paper aims to generate **customized, high-quality literature review paragraphs** that not only cite papers but **explain relationships** between them in a narrative style. Unlike earlier works that generated one citation sentence in isolation, this method generates **coherent paragraphs** with **transitions and exposition**.

Feature-based prompting for LLMs to extract:

- **Faceted summaries** (objective, method, findings, contributions, keywords)
- **Relationships between paper pairs**
- **Citation intent and usage** (dominant vs. reference type)
- **Contextual features** from the **target paper** (title, abstract, intro, conclusion)

Main idea plan:

- A short summary of the high-level ideas guiding the organization of the review.

Iterative generation and refinement:

- Initial paragraph generation.
- Enhance with **Cited Text Spans (CTS)** using ROUGE similarity scoring.

Implementation Pipeline

✅ **Step-by-Step Components**

1. Data Collection and Parsing

- Use **Google Search API** to find PDFs of cited papers.
- Convert PDFs to JSON using **doc2json** (Semantic Scholar's tool).
- Extract **title, abstract, introduction, and conclusion (TAIC)** from target paper.

2. Feature Extraction with LLMs (e.g., GPT-3.5-turbo)

- Generate **faceted summaries** for each cited and citing paper using TAIC.
- Extract **citation spans** using a **citation tagger** (Li et al., 2022).
- Summarize **relationships between paper pairs**.
- Generate **citation intent and usage** (dominant/reference, intent of citation).

3. Main Idea Generation

- From the related work section of the target paper (or feed topic), LLM generates a short **"main ideas"** plan to guide paragraph generation.

4. Paragraph Generation with GPT-4

- Use a structured prompt including:
    - Target paper's TAIC
    - Main idea plan
    - Faceted summaries
    - Relationships
    - Citation usage
- Generate **1-3 paragraph literature review**.

5. Cited Text Span Retrieval (CTS)

- Use ROUGE-1 and -2 recall between generated citation sentences and cited paper text to find **top-k sentences**.
- Regenerate the paragraph with CTS to improve detail and factual accuracy.

Human Expert Evaluation on 27 papers in NLP and adjacent fields:

- Scored on 8 criteria: fluency, coherence, relevance, factuality, usefulness, writing style, overall quality.
- **Best performance** when using **all features** and a **main idea plan**.
- Human judges disliked raw summaries—**preferred integrative, narrative-style writing**.

ROUGE Scores:

- Best models scored ~0.51 (ROUGE-1), 0.22 (ROUGE-2), 0.25 (ROUGE-L).

No fine-tuning — pure zero-shot prompting using:

- `gpt-3.5-turbo-0301` for feature extraction
- `gpt-4-0314` for literature review generation

Citation Tagger from Li et al. (2022) used to extract citation spans

doc2json used to parse papers from PDF

Faceted Summary Extraction for each paper
Title: {{title}}
Abstract: {{abstract}}
Introduction: {{introduction}}
Conclusion: {{conclusion}}
What are the objective, method, findings, contributions and keywords of the paper above?
Output Format:
Objective: ...
Method: ...
Findings: ...
Contribution: ...
Keywords: ...

B. **Relationship Between Paper Pairs**
Faceted summary of {{Paper A}}
Faceted summary of {{Paper B}}

Citation contexts:
1. {{sentence from A citing B}}
...

Very briefly explain the relationship between A and B. TLDR:
C. Enriched Citation Usage
How other papers cite {{Paper B}}:
- Relation between A1 and B
- Relation between A2 and B

Example citation fragments:
1. "..."
2. "..."

TLDR: {{Paper B}} is known for ... and cited for ...

D. Main Idea Plan of Target Paper
Human-written or LLM-generated summary of the target's related work section.
Prompt Format:
Write a short summary of the main idea of the following related work section. Ignore citations.
{{Related Work Section Text}}

3 Citation Paragraph Generation-
We have written our paper's TAIC:
Title: {{title}}
...
Main idea of our related work section:
{{main idea}}

List of cited papers:
1. {{Paper Title}} by {{Authors}}
{{Faceted Summary}}
<Usage> {{Citation Usage}}
How other papers cite it:

- {{Relation 1}}
- {{Relation 2}}
...

Generate a related work section (max 3 paragraphs), natural and concise.

LLAssist: Simple Tools for Automating Literature Review Using Large Language Models"1 introduces LLAssist, an open-source tool designed to streamline literature reviews in academic research by leveraging Large Language Models (LLMs) and Natural Language Processing (NLP) techniques-**LLAssist is presented as an open-source tool designed to streamline literature reviews in academic research**
The goal is to significantly reduce the time and effort required for comprehensive literature reviews, allowing researchers to focus on analysis and synthesis

Methodologically-LLAssist takes a CSV file of article metadata and abstracts, and a text file of research questions as input . For each article, it performs key semantics extraction (topics, entities, keywords) and relevance estimation based on binary relevance and contribution decisions (with scores), and provides reasoning for these assessments.... It then determines if an article is a "must-read" based on these scores and generates output in JSON and CSV formats . The tool uses a simulated Chain-of-Thought (CoT) prompting technique to enhance reasoning
LLAssist takes a CSV file of article metadata and abstracts, and a text file of research questions as input5 . For each article, it performs several key steps6 :

- **Key Semantics Extraction:** The tool extracts topics, entities, and keywords from the article's title and abstract using an extraction prompt sent to the LLM6 .
- **Relevance Estimation**: For each research question, LLAssist estimates the article's relevance ...:
  - **Binary Relevance Decision and Score**: A TRUE/FALSE value and a score (0-1) indicating the alignment with the research question. An article is considered relevant if its score exceeds 0.7, a threshold that can be adjusted .

Binary Contribution Decision and Score: A TRUE/FALSE value and a score (0-1) assessing the article's potential contribution to answering the research question. An article is considered contributing if its score exceeds 0.77 .

◦ **Relevance Reasoning**: A brief explanation of why the article is considered relevant7 .

◦ **Contribution Reasoning**: A justification for the estimated contribution7 .

• Must-Read Determination: Based on the relevance and contribution scores across all research questions, LLAssist determines if an article is a "must-read" using a logical OR operation on the thresholds.

• **Output Generation**: LLAssist provides output in JSON and CSV formats, including extracted semantics, relevance scores, and reasoning. This is intentionally designed to ensure human oversight in the process .

LLAssist can effectively identify relevant papers and works with various LLM backends . Different LLMs showed varying performance in relevance scoring and classification

LLAssist's performance in the field of LLM applications in cybersecurity using datasets from **IEEE Xplore and Scopus**

The evaluation assessed consistency, accuracy in matching papers to research questions, and the meaningfulness of reasoning, using different LLM backends: **Llama 3:8B, Gemma 2:9B, GPT-3.5-turbo-0125, and GPT-4o-**

LLAssist can effectively identify relevant papers and works with various LLM backends.

GPT-4o was the slowest but potentially more selective, while Llama 3 was the fastest but showed inconsistencies .... Gemma 2 showed reasonable performance and the potential for local use without cloud costs....

ChatCite, an LLM agent with human workflow guidance for comparative literature summary -

Reflective Incremental Generator module iteratively generates literature summaries- The Reflective Incremental Mechanism is a structured approach where an LLM

generates content in stages, producing multiple candidates at each step. A reflective evaluator (often another LLM) votes on these candidates, and top-scoring ones are retained for further expansion. This iterative, breadth-first-like process ensures higher quality, coherence, and stability in the final output.

Comparative Summarizer-In each iteration, it generates comparative analysis summaries by considering the relationship between the current reference paper, the proposed work, and previously summarized works. aims to enhance **comparability and organization** by providing **specific guidance (prompts)** to the LL**M.**

Reflective Mechanism: Uses an LLM as a Reflective Evaluator to vote on multiple candidate summaries per turn. Top-scoring candidates, selected via statistical analysis, are retained for the next round—ensuring quality and stability through a breadth-first-like approach.

To evaluate the quality of the generated summaries, the paper introduces a new LLM-based automatic evaluation metric called **G-Score**. This metric is based on **six evaluation** methods from human studies on literature reviews: **Consistency, Coherence, Comparative, Integrity, Fluency, and Cite Accuracy**

Key Element Extractor and the Comparative Incremental Mechanism significantly contribute to the improved performance of ChatCite in terms of ROUGE metrics and LLM-based evaluation metrics

In this paper the experimental results on the **NudtRwG-Citation dataset**. ChatCite, using GPT-3.5 as the decoder, was compared against baseline models including GPT-3.5 and GPT-4.0 under zero-shot and few-shot settings, as well as LitLLM with GPT-4.0

LLMs with human workflow guidance, as implemented in ChatCite, have the ability to effectively perform comprehensive comparative summarization of multiple documents