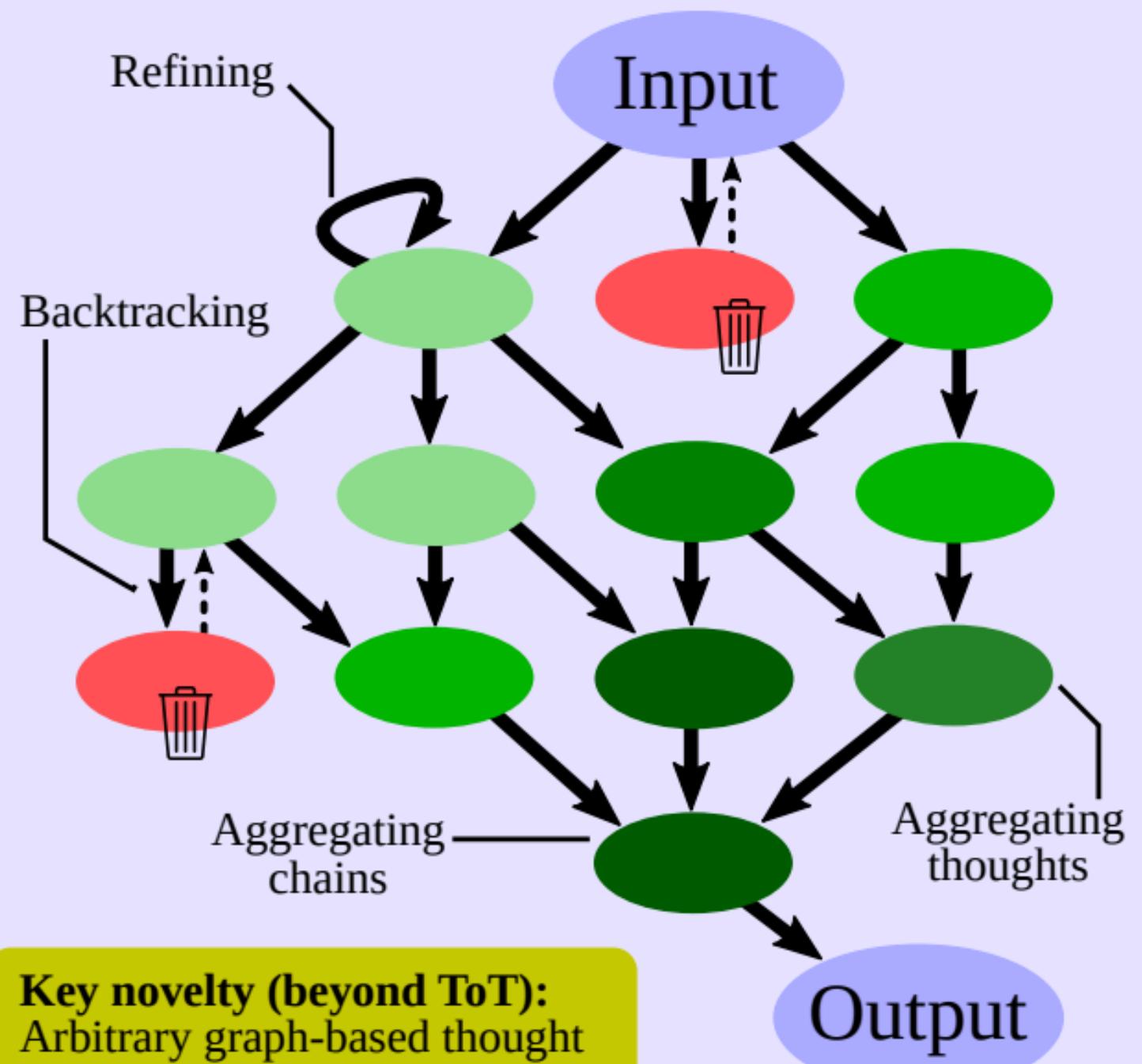


Graph of Thoughts (GoT)

[This work]



Key novelty (beyond ToT):
Arbitrary graph-based thought transformations (aggregating thoughts into a new one, looping over a thought to refine it)

ABOUT GRAPH OF THOUGHTS

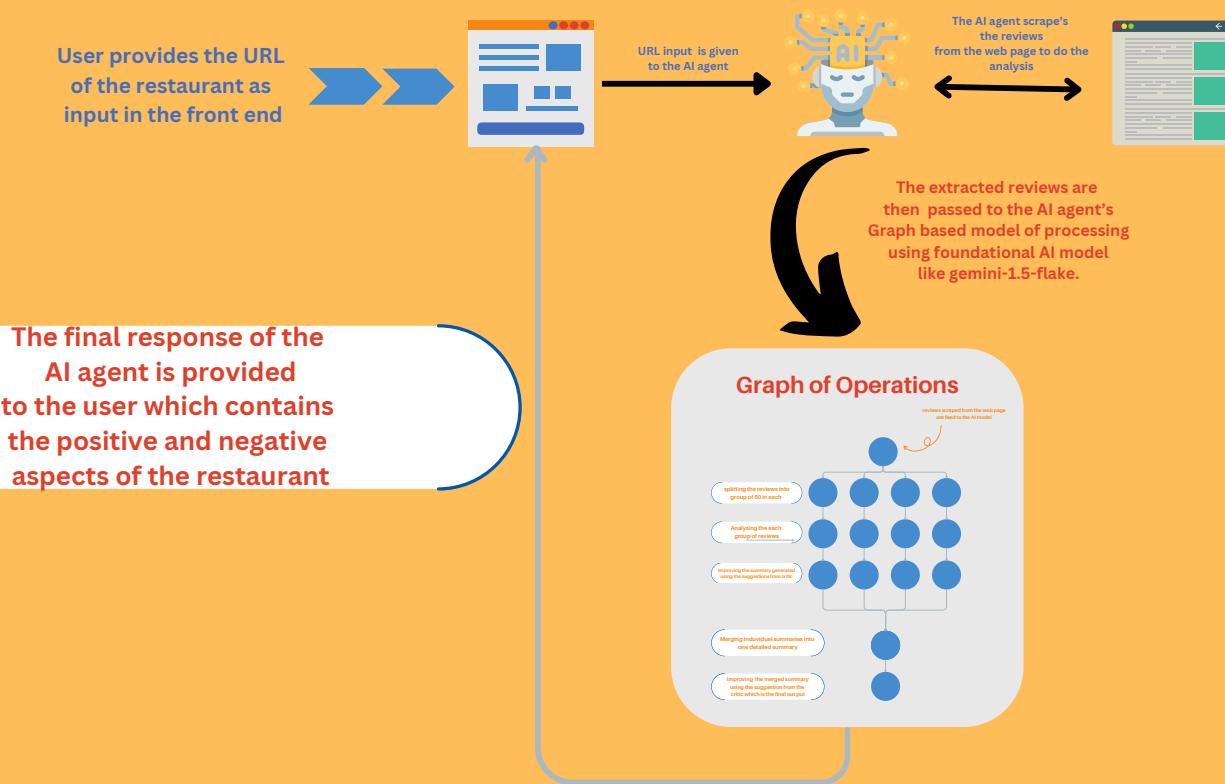
The Graph of Thoughts (GoT) is an AI reasoning framework that builds upon the Tree of Thoughts (ToT) technique, emphasizing non-linear exploration of ideas and solutions. In GoT, thoughts are represented as nodes in a graph, where edges denote possible transitions or relationships between them. Unlike tree structures, which have a rigid hierarchical flow, GoT allows for more dynamic exploration by enabling the AI to revisit and reevaluate previous thoughts, fostering a more flexible and interconnected reasoning process. This graph-based approach captures the complexity of problem-solving, making it possible to explore various pathways, link different ideas, and derive more nuanced insights.

By utilising GoT, the AI agent can better understand complex, multi-faceted problems by analyzing not only sequential steps but also the relationships between different ideas or solutions. This framework is especially useful for tasks like review analysis, creative problem-solving, or decision-making, where multiple variables and perspectives need to be considered. GoT's non-linear nature helps the AI arrive at more robust conclusions by revisiting earlier decisions, comparing alternative paths, and integrating insights from different directions within the graph.

ABOUT THE PROJECT

- Restaurant Review Scraping and Analysis: The AI agent takes a restaurant's URL as input, scrapes customer reviews from the site, and processes them for sentiment analysis. This data is then used as the foundation for deeper analysis using the Graph of Thoughts technique.
- Graph of Thoughts (GoT) for Enhanced Review Understanding: The AI agent structures its analysis by organizing the reviews into a "graph" of connected thoughts. Each thought or sentiment expressed in the reviews is treated as a node, and the AI explores different pathways and connections to understand trends, customer satisfaction, and key insights, leading to a more comprehensive and multi-faceted review analysis.

The complete flow chart of the system



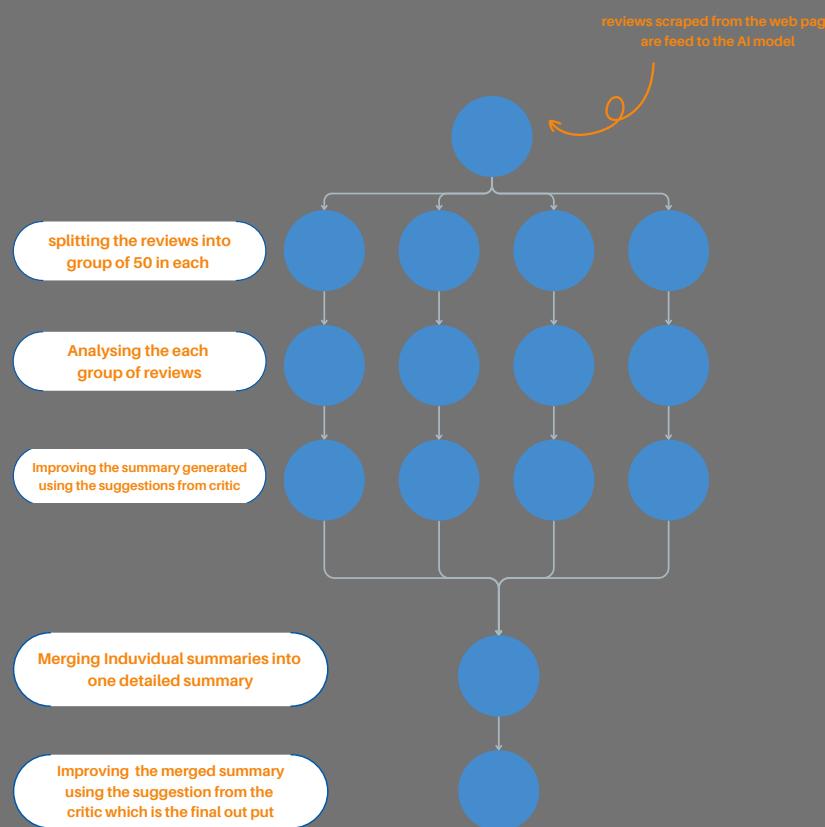
ABOUT THE TECHNOLOGY USED

The application consists of a React-based front end and a Node.js based backend. On the front end, users input the URL of a restaurant's review page, which is then submitted via an API request to the backend. This front-end interface ensures ease of use and seamless interaction for the user. Once the URL is submitted, the backend handles the core functionality, including review scraping and analysis.

The backend, built on Node.js, acts as the AI agent powered by the Gemini-1.5-flake model. It first uses the Cheerio module to scrape reviews from the restaurant webpage, extracting relevant data from the HTML content. After gathering the reviews, the AI employs the Graph of Thoughts (GoT) technique to analyze the sentiments and insights within the reviews. The agent follows the predefined **GoA** for analysing. This structured reasoning process allows the AI to explore different aspects of the reviews, providing a comprehensive and intelligent analysis.

WHAT IS GOA

The Graph of Operation (GoA)



In GoT the user/developer constructs a GoO instance, which prescribes the execution plan of thought operations. The GoO is a static structure that is constructed once, before the execution starts. Each operation object knows its predecessor and successor operations. Then, during the execution, an instance of the GRS maintains the continually updated information about the LLM reasoning process

The Graph of Operations (GoA) in our Graph of Thoughts (GoT) framework orchestrates a systematic and multi-step analysis of restaurant reviews to generate a comprehensive summary. Initially, the AI agent scrapes 200 reviews from the provided restaurant URL and divides them into four equal parts, each containing 50 reviews. Each subset is individually analyzed by the LLM model to extract positive and negative aspects, producing a preliminary summary. This summary is then evaluated by the same LLM model to ensure that all major points affecting user experience are covered. If any significant aspects are missing, the model provides suggestions for improvement, which are incorporated to refine the summary. After processing all four subsets through this iterative cycle, the four refined summaries are merged into a single final summary. This final merge is again assessed by the LLM model to verify that it encompasses all important points without redundancy. If gaps or overlaps are identified, the model offers further suggestions, and the final summary is adjusted accordingly. This structured operation flow ensures that the final output is both thorough and cohesive, effectively capturing the key insights from the restaurant reviews.

ALL AVAILABLE PROMPTS

- **getSummaryPrompt** : This prompt takes the reviews and consult the LLM model to get the detailed summary covering positive , negative aspects and also some suggestions.
- **scoreSummaryPrompt** : This prompt takes both reviews and the summary generated and scores it based on the amount of information cover which effect the user experience and amount of information redundant.If any major points are missing in the summary it also provides suggestions for improvement.
- **improveSummaryPrompt** : This prompt takes the reviews ,summary generated and the improvements suggested by the scoreSummaryPrompt and consult the LLM to improves the summary based on the suggestions made by reassessing both summary and the reviews
- **mergeSummaryPrompt** : This prompt takes the 4 sub summaries generated and consult the LLM to merge them into one final summary which covers all the points in the sub summaries with out redundancy
- **scoreTheMergePrompt and improveTheMergePrompt** : scoreTheMergePrompt prompt takes the 4 sub summaries generated and the merged summary consult the LLM to score the merge based on redundancy and amount of information retrieved , it also makes suggestion if needed. Then the merged summary can be improved further based on the suggestions using improveTheMergePrompt.

ALL MODULES

- **scraper** : This module is responsible for scraping the review pages of the given restaurant then parse the HTML to extract the reviews and provide the reviews in array formate . It uses **cheer.IO** for this task.
- **getResponseInSections** : This module is responsible for dividing the reviews in to 4 sub sets and analyse each sets using the LLM and the getSummaryPrompt , then it uses the scoreSummaryPrompt to score each of the summaries and then further improves each summaries based on the suggestions made returns all the 4 improved summaries in a valid JSON formate
- **mergeAllSummaries**: This module is responsible merging all the four summaries in to one final summary using the mergePrompt.It then scores the merged summary and then improves it if needed by using scoreTheMergePrompt and improveTheMergePrompt and finally returns the improved merged summary in valid JSON formate.



EXAMPLE OUT PUT

positivePoints:

- 1.The restaurant is praised for its authentic Rajasthani flavors and commitment to a cultural dining experience.
- 2.Customers consistently appreciate the politeness and attentiveness of the staff, creating a welcoming atmosphere.
- 3.The food is generally considered reasonably priced, especially the thalis which offer a diverse selection.
- 4.Customers frequently praised the delicious taste of the food, particularly the dal bhati, churma, and aamras.
- 5.The packaging is consistently praised, described as impressive, royal, and a standout feature.

negativePoints:

- 1.Multiple customers expressed dissatisfaction with inconsistent food quality, reporting issues like blandness, stale food, and raw ingredients
- 2.Hygiene concerns are a recurring issue, with reports of dirty utensils, leaking AC, unhygienic conditions in the washrooms and food preparation areas.
- 3.Several customers mentioned issues with service, including slow service, inattentive staff, difficulty paying, particularly with Swiggy Dineout, and lack of availability of high chairs for children.
- 4.Customers have expressed concerns about portion sizes, finding them smaller than expected and not value for money. The disparity between the food presentation in pictures and the actual portions served is also a concern.
- 5.Concerns were raised regarding the quality of some dishes, including cold poori, bad-tasting gatta sabji and kadhi, unpromising dal & baati, and even rotten rice.

Suggestions:

- 1.Implement consistent quality control measures to ensure consistently good food quality and address concerns about stale or chilled food. Addressing the recurring health issues reported by customers is crucial.
- 2.Prioritize the maintenance and cleanliness of the washrooms and all areas, including food preparation, to ensure customer comfort and trust.
- 3.Invest in staff training to enhance service quality and address customer needs more effectively, including handling birthday celebrations and addressing concerns about cash-only payments.
- 4.Review portion sizes to ensure they align with customer expectations and price points, particularly focusing on providing sufficient amounts of dal, sabzi, and batni.
- 5.Actively engage with customer feedback, addressing concerns and demonstrating a genuine commitment to improving the overall experience.



THANK YOU