WEB-Cognize AI enhanced Information synthesis

Author1: **Abhay Nosran**
Department of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India
abhay.21scse1300026@galgotiasunive rsity.edu.in

Author2: **Divyansh Saini**
Department of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India
divyansh.21scse1011361@galgoti asuniversity.edu.in

Author3: **Gurpreet singh**
Department of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India
gurpreetsingh@galgotiasuniversity.edu.in

Abstract—
Finding Information and keeping its integrity is a difficult cognitive process that takes a lot of time and energy. Even while search engines have revolutionized the way we obtain information, they frequently fail to capture complex human intents. To address this issue, recent developments in large language model s (LARGE LANGAUGE MODEL s) have sparked attempts to integrate LARGE LANGAUGE MODEL s with search engines. However, there are three main challenges to these strategies: (1) It is not always possible to fully and accurately retrieve complex queries in a single search; (2) pertinent information is frequently separated from irrelevant content on multiple web pages; and (3) the volume of web pages with extensive content can quickly surpass the context length limitations of LARGE LANGAUGE MODEL s. We suggest WEB-Cognize, a framework for web-based information integration and searching that is model ed after human cognitive processes, to overcome these problems. Using a multi-agent LARGE LANGAUGE MODEL -based architecture, this system consists of two parts: Web planner and Web Searcher. By segmenting a user query into smaller subquestions and arranging them in a dynamic graph structure, the Web planner mimics human multi-step thinking. This graph is repeatedly expanded using the Web Searcher's findings. In turn, the Web Searcher retrieves pertinent data from search engines in a hierarchical manner, obtaining useful knowledge to improve the Web planner's procedure. . WEB-Cognize's multiagent architecture allows for the quick and effective parallel processing of vast amounts of data (such as more than 300 web pages) in a short amount of time, which is comparable to finishing jobs that would normally take hours for people. For both closed-set and open-set inquiries.

I. INTRODUCTION

An important cognitive function that supports analysis and decision-making in a variety of fields is the search for and integration of information. Nevertheless, this procedure frequently requires a significant amount of time and work. Although information retrieval has been transformed and made simpler by the introduction of search engines (Brin & Page, 1998; Berkhin, 2005), there are still difficulties in integrating web-based information to fulfill complex human intentions. Recent developments in Large Language Models (LARGE LANGAUGE MODEL s) have shown impressive ability in a variety of domains, including reasoning, language comprehension, and information synthesis.

A special potential for their integration is presented by the complementing qualities of search engines and LARGE LANGAUGE MODEL s. While search engines provide users access to enormous databases of online information, LARGE LANGAUGE MODEL s are superior at thinking and understanding. Research acquisation and integration procedures might be revolutionized by combining these techniques. This issue is frequently handled as a simple retrieve-augmented generation (RAG) problem in traditional ways. However, the depth and complexity needed for complicated user requests are sometimes too much for this straightforward architecture to handle. It faces three main difficulties:

1. Complex Problem Decomposition: In order to retrieve pertinent information, real-world inquiries frequently require in-depth analysis and the division of the problem into smaller parts, which is not possible when retrieving web pages all at once.
2. Overwhelming Information Noise: LARGE LANGAUGE MODEL s' capacity to effectively incorporate pertinent content is
hampered by the enormous volume of retrieved web pages as well as irrelevant information.
3. Context Length Limitations: Integration performance may suffer if the quick accumulation of retrieved material exceeds the context length limitations of LARGE LANGAUGE MODEL s. Motivated by the manner in which human specialists tackle intricate problems, we present WEB-Cognize, a simplified yet potent multi-agent framework founded on LARGE LANGAUGE MODEL s. Web planner and Web Searcher are the two main components of this system. By dividing a user query into more manageable sub-questions and allocating them to appropriate Web Searchers, the Web planner simulates human cognitive thinking. Web planner describes this process as an iterative graph building to further improve problem-solving. By adding nodes and edges to a topological graph representation, it dynamically breaks down the query into sequential or parallel sub-problems using established Python-based interfaces.

By using a hierarchical retrieval strategy, the Web Searcher manages distinct sub-problems and extracts valuable information from a large number of search results. This architecture minimizes the difficulties caused by massive amounts of online material while guaranteeing the effective aggregation of pertinent data. By dividing the task across several agents, WEB-Cognize lessens the strain on any one component, allowing for improved extended context management and bridging the gap between search engines' ability to retrieve raw data and LARGE LANGAUGE MODEL s' capacity for nuanced comprehension.

Using the GPT-4o and InternLM2.5-7B-Chat model s, we carried out comprehensive tests on both closed-set and open-set question-answering (QA) tasks to evaluate WEBCognize's efficacy. The quality of responses has significantly improved, according to the results, especially in terms of breadth and depth. Furthermore, WEBCognize's outputs were favored by human assessors above those produced by other platforms such as Perplexity Pro and ChatGPT-Web (GPT-4o). These results demonstrate WEBCognize's potential as a strong and competitive answer for AI-driven search engines, even when combined with open-source LARGE LANGAUGE MODEL s.

2 WEB-Cognize

A Web planner and a team of Web Searchers make up WEB-Cognize, which aims to efficiently combine the
reasoning and information integration skills of LARGE LANGAUGE MODEL s with the web information retrieval capabilities of search engines. Using graph reasoning, Web planner first breaks down the user query into sequential or parallel
search tasks before deciding on the next course of action depending on the search feedback. In order to
respond to sub-questions, Web Searcher is given the query and uses hierarchical information retrieval on the
Internet. In the context of the multi-agent design, we also go over context management.

2.1 **Web planner**: Planning via Graph Construction

As a high-level planner, the Web planner coordinates other agents and orchestrates the reasoning stages.
However, we found that speed is not adequate when the LARGE LANGAUGE MODEL is just prompted to plan the full data pipeline
architecture. In particular, existing LARGE LANGAUGE MODEL s produce coarse-grained search queries because they have trouble breaking down complicated topics and comprehending their topological links. This method underutilizes LARGE LANGAUGE MODEL s' ability to provide as a bridge between people and search engines, translating human intents into detailed search tasks and providing precise results.

E, where V is a collection of nodes v, each of which represents a separate web search. The END node represents the final response, while the auxiliary START node represents the beginning question. E stands for directed edges, which show the topological links between nodes in reasoning (search contents). This DAG formalism gives LARGE LANGAUGE MODEL s a more formal and understandable representation while capturing the intricacy of determining the best execution path. We use code writing to actively urge the model to interact with the graph, taking advantage of the current LARGE LANGAUGE MODEL s' greater performance on code jobs. To do this, we added nodes or edges to the network using predefined atomic code methods. Using a Python interpreter, the LARGE LANGAUGE MODEL generates new code and concepts for reasoning on the mind graph after reading the complete discourse at each turn, including previously created code and online search results. When a node is added to the

reasoning graph during execution, a Web Searcher is called to carry out the search and compile the data. Given that We can parallel the newly added nodes to attain a significantly quicker information aggregation speed because they are only reliant on nodes created in earlier rounds. The planner adds the end node to the final answer once all the data has been gathered. Web planner dynamically constructs the reasoning path by interacting with the graph through unified code operations through integration with the Python interpreter. The LARGE LANGAUGE MODEL can fully utilize its superior code generation capabilities thanks to this "code as planning" method, which improves control and data flow in long-context scenarios and improves performance when tackling complicated issues. 2.2 Web Searcher: Hierarchical Retrieval for Web Browsing.

2.2 Web Searcher summarizes useful replies based on search results, functioning as an advanced RAG (Retrieve-and-Generate) agent with internet connectivity. LARGE LANGAUGE MODEL s find it difficult to evaluate all linked pages within a certain context length because of the vast amount of material available on the internet. We use a simple coarse-to-fine selection approach to deal with this. In order to expand the search content and enhance the memory of pertinent information, the LARGE LANGAUGE MODEL first creates a number of related queries based on the Web Planner's provided questions. LARGE LANGAUGE MODEL uses the search results to create a response that addresses the initial query. This hierarchical retrieval technique makes it much easier to navigate large web pages and enables the effective extraction of highly relevant material with detailed information. A detailed working pipeline of Web Searcher.

2.3 LARGE LANGAUGE MODEL Context Management in WEB-Cognize A straightforward multi-agent solution for intricate information searching and search engine integration is offered by WEB-Cognize. Long-context management across several agents is also automatically made possible by such a paradigm, which enhances the framework's overall effectiveness, particularly in situations when the model must swiftly scan a large number of web pages. Web planner can concentrate entirely on the deconstruction and analysis of the user inquiry without being sidetracked by the lengthy online search results because it divides the search responsibilities among distinct search agents and only uses the Web Searcher's searched results. Without being distracted by other contents, each Web Searcher simply has to look for the contents that match its assigned sub-query. Because of the clear role distributionWEB-Cognize provides aeffective context management solution for long-context activities for LARGE LANGAUGE MODEL by significantly reducing context computation during the whole procedure. For training single LARGE LANGAUGE MODEL s, such a multi-agent architecture also offers a clear-cut and uncomplicated long-context task building pipeline. In the end, WEBCognize gathers and combines relevant data from over 300 pages in less than three minutes, whereas a comparable cognitive task may take human professionals over three hours to complete. The explicit context state transfer across several agents necessitates careful handling of the context throughout the process. Because of the small receptive field within the search agent, we experimentally discover that just concentrating the deconstructed question from the Planner may result in the loss of important information during the information gathering phase. It is difficult to manage context amongst several agents in an efficient manner.

We discover that we can manage the context across several agents with ease thanks to the topological relations that are created by the directed graph edges. More precisely, when we run each search agent, we just prefix the answer from both the root node and its father node. As a result, each Web Searcher may efficiently concentrate on its subtask without losing sight of the ultimate objective or the previously relevant context.

3 Experiments:

In order to assess WEB-Cognize's subjective and objective performance, we use two main types of Question Answering To provide a fair comparison, no additional reference sources are taken into account, and all model s only have access to the Internet via the BING search API. or chat gpt 4 3.1. Results and Analysis The evaluation's findings are shown, and we also offer numerical findings in The superiority of our suggested Web planner is shown by Figure, which shows a complete improvement in the breadth and depth of the model response. In order to balance the trade-off between time efficiency and search space exploration, LARGE LANGAUGE MODEL is able to gradually break down the complicated problem into executable queries by including code into the DAG creation step. In addition, compared to previous model s, WEB-Cognize provides more succinct and thorough answers by going over more specific search terms related to the query. But Better facticity performance is not achieved by WEB-Cognize. We believe that more specific search results might divert the model 's focus from the original issue, particularly when LARGE LANGAUGE MODEL has limited long-context capabilities. Therefore, addressing hallucination problems while online browsing is a logical next step for online-Cognize. A screenshot of working model with a simple query 4. Related Work 4.1 Tool Utilization with LARGE LANGAUGE MODEL Technical Report 5 With the use of the Tool Learning framework, LARGE LANGAUGE MODEL s may easily interface with a range of tools, including databases, search engines, and APIs, providing dynamic answers to challenging issues. In addition to boosting LARGE LANGAUGE MODEL s' interpretability and reliability, this integration increases their resilience and flexibility to a variety of tasks, such as question answering and lowering hallucination code creation. Enhancing Tool Learning systems' tool integration component has been the subject of recent study.

To guarantee that LARGE LANGAUGE MODEL s can obtain the most relevant tools for a job, works like these have focused on enhancing the retrieval procedures. In order to optimize

the reading and comprehension processes inside the framework, further research seek to improve the LARGE LANGAUGE MODEL s' capacity to use the material that has been retrieved.

4.2 RAG with LARGE LANGAUGE MODEL With the incorporation of search engines, RAG shows notable benefits in solving knowledge intensive issues, particularly in open-domain settings. By integrating with the retriever, RAG enables LARGE LANGAUGE MODEL s to deliver efficient solutions and timely information. Additionally, RAG is used for a variety of activities including question answering, code generation, and hallucination reduction. Recent research has concentrated on improving RAG systems' retrieval component, while other studies have improved the language model 's capacity to optimize the framework as a reader. As LARGE LANGAUGE MODEL capabilities have improved, several researchers have started to revamp model training approaches and optimize frameworks. LARGE LANGAUGE MODEL is trained by SAIL to concentrate more on reliable and instructive search results. LMMs can autonomously retrieve, analyze, and enhance their text generating skills thanks to self-RAG. By learning to improve queries through an iterative approach, improves query formulation. By accessing useful information from the Internet, our approach improves answer quality by integrating web search capabilities into LARGE LANGAUGE MODEL

4.3 Web Agents From simple tools that could answer questions to complicated systems that could perform intricate online interactions, web automation agents have seen significant development. QA activities were the main focus of early model s like Web GPT and Web GLM, but more dynamic processes are now the focus of more recent developments. This development is shown by MindAct and Web Agent, the latter of which, in spite of deployment issues brought on by its scale, exhibits outstanding web navigation. Auto Web GLM provides a sensible substitute with stronger features and a smaller model size. The area is advancing toward scalable and adaptable solutions for real-world applications as reinforcement learning and behavior cloning techniques are used to enable ever more autonomous and effective online automation. This study employs a multi-agent architecture to address the primary problems and focuses more on the task of web research acquisation and search engine integration than on web surfing.

4 Conclusion:

By more fully using the advantages of search engines for complicated online information searching and authenticating tasks. By representing the problem-solving process as graph construction, WEB-Cognize performs efficient and adequate decomposition of complicated queries to enhance the accuracy and memory of the obtained pertinent online information. Specialized agents share the cognitive load in the multi-agent design enabling strong management and protracted settings. WEB- Cognize' s competitive advantage in AI-driven search solutions is demonstrated by the findings that human evaluators favored its replies above those of ChatGPT-Web and Perplexity.ai. Future studies on multi-agent frameworks for handling complicated cognitive tasks at the human level are anticipated to be facilitated by this study.

References

- Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 30(1-7), 107-117. https://doi.org/10.1016/S0169-7552(98)00110-X
- Berkhin, P. (2005). A survey on PageRank computing. Internet Mathematics, 2(1), 73-120. https://doi.org/10.1080/15427951.2005.10129093
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language model s are few-shot learners. Advances in Neural Information Processing Systems, 33, 18771901.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language model s are unsupervised multitask learners. OpenAI Blog, 1(8), 9.
- Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language model s trained on code. arXiv preprint arXiv:2107.03374.
- Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., ... & Goyal, S. (2021). Improving language model s by retrieving from trillions of tokens. arXiv preprint arXiv:2112.04426.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1-67.
- Xie, Q., Luong, M. T., Hovy, E., & Le, Q. V. (2020). Selftraining with noisy student improves ImageNet classification. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10687-10698. https://doi.org/10.1109/CVPR42600.2020.01070
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation model s. arXiv preprint arXiv:2108.07258.
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... & Yih, W. T. (2020). Dense passage retrieval for open-domain question answering. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 67696781. https://doi.org/10.18653/v1/2020.emnlp-main.550
- Ni, J., Urbanek, J., Sukhbaatar, S., Fan, A., & Weston, J. (2022). Learning to retrieve from Web-scale knowledge with memory networks. Proceedings of the 10th International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.2112.02133
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). Retrievalaugmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459-9474