Fact-Centric Knowledge Web for Information Retrieval

Reuben Sinha
Department of Information Technology
Nagoya Institute of Technology
Nagoya, Japan
reubensinha7@gmail.com

Shun Shiramatsu
Department of Information Technology
Nagoya Institute of Technology
Nagoya, Japan
siramatu@nitech.ac.jp

***Abstract*—The unreliability of Large Language Models (LLMs) owing to Machine Hallucination necessitates a shift towards Retrieval Augmented Generation (RAG), to ensure the veracity of LLM generated content. Inspired by Knowledge Graph based RAG techniques and their advantages of reduced token cost, improved computational performance and knowledge discovery, we present a "Knowledge Web" Structure and its associated techniques for Information Storage and Retrieval. We also explore the structure's synergy with a Fact Finding LLM Agent, capable of generating answers for multi-hop questions from challenging datasets such as the MuSiQue dataset. While still in the experimental phase, our initial results are promising, indicating the potential efficacy of our approach compared to existing RAG techniques.**

***Keywords*—Knowledge Graph, Large Language Models, LLM Agent, Information Storage, Information Retrieval**

I. Introduction

The growing context lengths in Large Language Models (LLM), coupled with extremely expensive training costs have pivoted the LLM ecosystem into Retrieval Augmented Generation (RAG) [1], encompassing an array of techniques, out of which Knowledge Graph has been observed to provide improved efficiency in cost as well as performance owing to its graphical structure [2]. Knowledge Graphs are also a viable solution to the "Lost in the Middle" problem where the position of relevant information within large contexts effects the quality of generated content [3]. A knowledge graph can potentially eliminate the Lost in the Middle problem as it facilitates reduced token usage. However, despite these advantages, the process of converting text into a knowledge graph often results in loss of information when the text data is stripped and reduced into an ordered triplet to generate a knowledge graph, further restricting knowledge graph-based RAGs to use-cases with affinity for the ordered-triplet structure of traditional knowledge graphs.

Aiming to overcome this limitation and achieve a domain-unrestricted Knowledge Graph based RAG, we present a Fact-Centric Knowledge Web, consisting of keyword-indexed web of context independent facts, decomposed from text documents with minimal information loss using LLM. We also explore associated techniques of information retrieval by leveraging vector indices derived from keywords, and a fact ranking score criterion to retrieve highly relevant information from the Knowledge Web. The Knowledge Web also synergizes with self-prompting AI Agents powered by LLM's Chain of Thought (CoT) [4], facilitating Autonomous Knowledge Discovery, enabling multi-hop question answering.

This work was partially supported by JST CREST (JPMJCR20D1), NEDO (JPNP20006), and JSPS KAKANHI (24K03052, 22K12325).

II. Structure of Knowledge Web

Inspired by the Knowledge Graph, the Knowledge Web is largely a graphical structure consisting of context independent facts, decomposed from documents, and interconnected through keywords which also act as indices. Unlike traditional knowledge graphs consisting of ordered triplets to represent entities and relations between them, our approach generates a Knowledge Web founded on a relatively simple ordered pair structure at its core, by establishing relationships between facts and keywords.

Figure 1, contrasts the structures of Knowledge Web and a traditional Knowledge Graph. The first structure depicts a Knowledge Web where, "Barack Obama", "United States of America" and "President" are keywords, associated to the fact which expresses the context between all three keywords. Such a structure, unlike the ordered triplets of entities in traditional Knowledge Graphs, ensures minimal loss of information by preventing the trimming of text data. The second structure in Figure 1 displays a traditional knowledge graph, where the period of presidency is omitted due to structural restrictions.

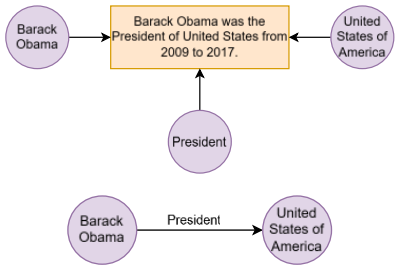

Fig. 1. An example contrasting the structure of Knowledge Web and a traditional Knowledge Graph respectively.

Additionally, every keyword in the Knowledge Web is associated with its context specific vector embeddings that facilitates effective information retrieval.

III. System Design

The system under study consists of two highly customizable components: Knowledge web generation component, and the Information retrieval component. LLMs play a crucial role in both components using the single-shot learning principle [5], making the system customizable, future-proof and adaptable even as the capabilities of LLMs improve.

A. Knowledge Web Generation

Figure 2 depicts the generation of the Fact-Centric Knowledge Web, which consists of 3 steps: 1. Vector indexing of Keywords, 2. Decomposition of documents into

979-8-3315-0494-6/24/$31.00 ©2024 IEEE
DOI 10.1109/WI-IAT62293.2024.00067

Facts and 3. Linking of Facts and Keywords. Unaltered LLMs are employed in all 3 steps, prompted with single-shot learning which can be altered to accommodate specific use-cases while maintaining proper functionality.

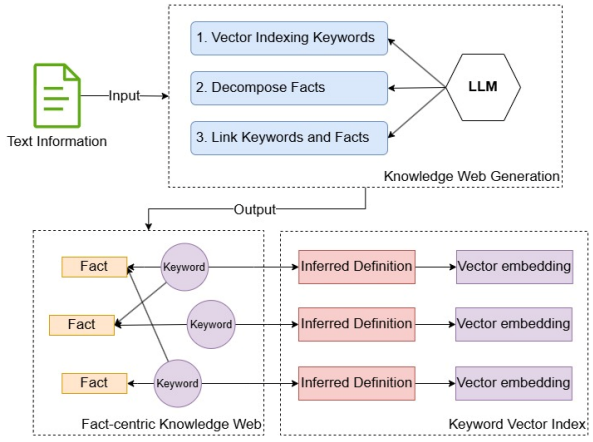

Fig. 2. An overview of the generation of Knowledge Web. A document is decomposed into facts, linked to Vector-Indexed Keywords.

a. Details about Vector Indexing of Keywords

Keywords are nouns or compound phrases which the LLM deems as essential entities within the uploaded document. With the aim of avoiding domain restriction by binding keywords to pre-determined entity types, we have chosen to prioritize their semantics by binding keywords to the vector embeddings of their "inferred definition" that are context-specific. In our observation, LLMs are capable of accurately generating context-specific definitions for the keywords in documents such that even lexically similar keywords are effectively delineated within the generated Knowledge Web.

For instance, in Table 1, the keyword "Helios" is stored as two unique nodes in the Knowledge Web, despite the lexical similarities between keywords as they semantically differ when considered within their respective contexts.

TABLE 1. Lexically equivalent, but semantically different.

Keyword	Simplified Definition
Helios	Name of a Sun God in Ancient Greece
Helios	Name of a Spacecraft

With the intent of leveraging vector embeddings of keywords for the retrieval operation, only a simplified definition is considered, as a query often contains far less context to generate a detailed definition for a keyword which is sufficiently similar to the embedding stored within the Knowledge Web.

b. Details about Decomposing Documents

Decomposition of documents into facts facilitates advantages associated with Knowledge Graphs, such as information reorganization, knowledge discovery and cost-effective information retrieval by limiting the retrieval of unnecessary information when queried [2]. This component utilizes an LLM at its core, functioning through single shot learning; Common concerns such as coreference resolution, context independence and managing temporal dependencies amongst generated facts are handled surprisingly well by the LLM. Table 2 provides an example of the document decomposition output generated by GPT-4o. In the example, the document has been decomposed into facts of minimal granularity. The increase in volume of text for the generated facts is an expected consequence and is often proportional to the number of facts generated to make them context independent to each. This granularity can be adjusted by modifying the single-shot example to suit a specific use case.

TABLE 2. Decomposition of documents into facts

Original Documents	Generated Facts
Green is the fourth studio album by British progressive rock musician Steve Hillage. Written in spring 1977 at the same time as his previous album, the funk-inflected "Motivation Radio" (1977), "Green" was originally going to be released as "The Green Album" as a companion to "The Red Album" (the originally intended name for "Motivation Radio"). However, this plan was dropped and after a US tour in late 1977, "Green" was recorded alone, primarily in Dorking, Surrey, and in London.	1. Green is a fourth studio album. 2. Green is by Steve Hillage. 3. Steve Hillage is a British musician. 4. Green is in the progressive rock genre. 5. Green was written in spring 1977. 6. Motivation Radio is a funk-inflected album. 7. Motivation Radio was released in 1977. 8. Green was originally going to be called The Green Album. 9. Motivation Radio was originally going to be called The Red Album. 10. The Green Album and The Red Album were intended to be companion albums. 11. Green was recorded after a US tour in late 1977. 12. Green was recorded in Dorking, Surrey, and London.

c. Details about linking keywords and facts

With the intent of using keywords as indices during the information retrieval process, our design advocates a dense graphical structure consisting of relationships between keywords, and all its associated facts. The system establishes two types of relationships between facts and keywords: Explicit and Implicit relationships. Explicit relationships are considered trivial and represents the presence of a keyword inside its associated fact. In contrast, an Implicit relationship is non-trivial and represents a contextual connection between a keyword and a fact, when the keyword is not explicitly present in the fact.

d. Details about determining Implicit relationships

Although the task of determining implicit relationships involves the use of LLM employed through a single shot example, our observation showed higher levels of inconsistency when the association between implicitly related keywords with their facts were directly extracted. Thus, our approach instead considers the relationship between keywords themselves. The keyword-to-keyword relationships capture the contextual relations between the keywords within documents, which are then relayed to link keywords implicitly to facts.

In Figure 3, "Avatar 2" and "Film" are keywords, which share a contextual relationship. Film has an explicit relationship with its fact. The contextual relationship between "Avatar 2" and "Film" is used to establish an implicit relationship between the fact and "Avatar 2".

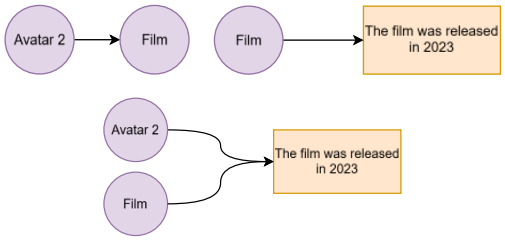

Fig. 3. Establishing implicit relationships between keywords and facts

B. *Information Retrieval*

Information retrieval is performed in tandem with the structure of the Knowledge Web to effectively respond to a query. Figure 4, depicts the two steps involved in identifying the relevant facts containing the required information: Keyword Node Resolution and Ranked Fact Retrieval.

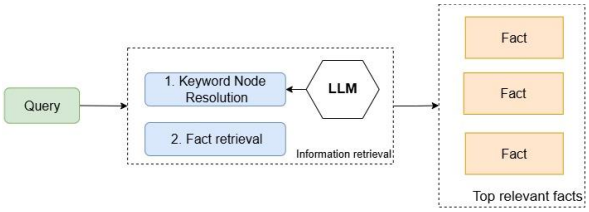

Fig. 4. An overview of information retrieval

Additionally, the information retrieval process is further enhanced through an LLM agent designed to self-prompt and further explore the Fact-Centric Knowledge Web to facilitate the answering of challenging multi-hop questions.

a. *Details about Keyword Resolution*

The keywords in a query include words and compound words, as well as their subsets, which are extracted using LLM. Amongst the extracted keywords, those present in the Fact-Centric Knowledge Web are retrieved using cosine similarity of vector embeddings of their simplified definitions. However, since simple definitions of keywords are utilized, the procured definitions for drastically different words, such as names of people can potentially be the same. In table 3, "Steve Hillage" and "Barack Obama" are simply defined as "Name of a person". Thus, a lexical similarity score, called the Keyword Resolution Score, a derivative of Levenshtein's distance (LD), provides the means to eliminate false positives in the Keyword Node Resolution process.

$$1 - \frac{\text{LD}(\text{query keyword, candidate keyword})}{\max\big(\text{len}(\text{query keyword}), \text{len}(\text{candidate keyword})\big)}$$

The query keyword is extracted from the query; the *candidate keyword* is retrieved from the vector database.

Table 3 highlights the necessity of the Keyword Resolution Score, where the pairs of keywords (Steve Hillage, Barack Obama), and (London, Dorking) share the same simplified definition, however their Keyword Resolution Scores with respect to each other are extremely low, thus dismissing their equivalence.

TABLE 3. Conflicting definitions of keywords

Keyword	Simplified Definition	Keyword resolution score
Steve Hillage	Name of a Person	0.07692
Barack Obama	Name of a Person	
London	A Place in England	0.28571
Dorking	A Place in England	

b. *Details about Fact Retrieval*

Figure 5 depicts the steps involved in the fact retrieval process. The facts connected to keywords extracted from the query are ranked and restricted prior to being retrieved, by using the query itself as a basis to determine relevancy.

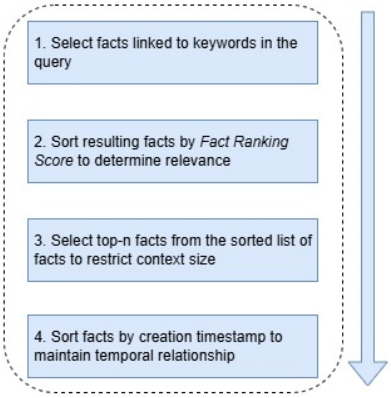

Fig. 5. An overview of Fact retrieval

The number of facts retrieved are restricted to reduce token usage, by accounting for the "hotkey problem" where a few popular keywords can have a large number of related facts, which if left unchecked, violates the intent of limiting context length downstream for the subsequent steps involving the LLM. A ranking algorithm consisting of the following formula for the Fact Ranking Score is used to select the top relevant facts.

$$\text{cosSim}(\text{query, fact}) + \text{degree}(\text{fact})$$

cosSim(query, fact) stands for the cosine similarity between the vector embeddings of the query and a fact; *degree(fact)* stands for the number of extracted keywords from the query related to the fact. Since *degree(fact)* is a whole number and *cosSim(query, fact)* is a fractional value, the Fact Ranking Score elegantly establishes a 2 Dimensional ranking mechanism, allowing facts to be sorted in non-increasing order by its *degree(fact)* and *cosSim(query, fact)* respectively.

Table 4 provides an example which highlights its inherent two dimensional sorting capability which contributes to determining a fact's relevancy to a query. This abides with the reasoning that relevancy of facts are proportional to the number of keywords from the query connected to it, followed by the its cosine similarity to the query itself. Lastly, the selected facts are ordered by non-decreasing creational timestamp, to maintain the temporal relationship amongst them.

TABLE 4. An example of fact retrieval scores ranked by non-increasing order of score

Fact ID	degree(fact)	cosSim(query, fact)	Score
214	3	0.88	3.88
216	3	0.72	3.72
312	1	0.91	1.91

c. *Details about Retrieval Agent*

LLM agents can utilize Chain of Thought to self-automate tasks [4]. Inspired by the success of Interleaving Retrieval with Chain of Though (IRCoT) in answering complex, multi-hop queries which require inference [6], our system utilizes a Fact Finding Agent with a custom algorithm to recursively perpetuate the fact retrieval process until the relevant information is retrieved. As depicted in Figure 6, the Agent recursively queries the knowledge web utilizing the retrieved facts to prepare the next query, enabling information gathering and multi-hop information retrieval.

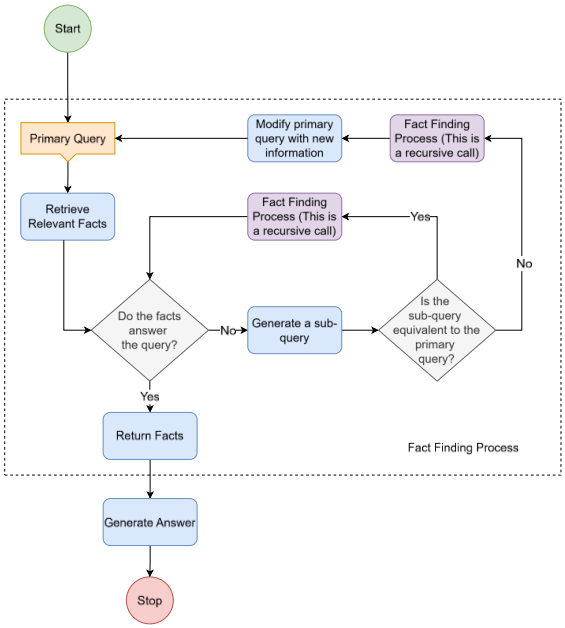

Fig. 6. Fact Finding Algorithm used by the LLM Agent

The algorithm for the Fact Finding Process used by the Agent constitutes the following:

1. Given the primary query, the system retrieves relevant facts from the Knowledge Web.
2. The LLM itself acts as a judge, to determine whether the facts retrieved are sufficient to generate a response. If the facts are sufficient, they are used to generate an answer. If not, a new sub-query perpetuates the search for a response.
3. If the sub-query is equivalent to the primary query, then the facts retrieved for the sub-query shall be used to generate an answer.
4. If the sub-query is not equivalent to the primary query, it implies that the sub-query answers only a portion of the primary query and additional searching is required. Thus the facts retrieved for the sub-query will be used to augment the primary query and the fact finding process will be repeated.

As an example, consider the following query from the MuSiQue dataset, "Who founded the political party of Dimuthu Bandara Abayakoon?". Tables 5 and 6, depicts the fact finding process according to the logs recorded by the system.

TABLE 5. Main Query and Facts retrieved

Query	Retrieved Facts
Who founded the political party of Dimuthu Bandara Abayakoon?	1. Dimuthu Bandara Abayakoon is a Sri Lankan politician. 2. Dimuthu Bandara Abayakoon belongs to the Janatha Vimukthi Peramuna. 3. Dimuthu Bandara Abayakoon was elected in the 2004 election.

Since the retrieved information is insufficient to generate the appropriate response, a sub-query generated is " Who founded the Janatha Vimukthi Peramuna?", which is equivalent to the primary query. Thus, the fact finding process is invoked again to answer the sub-query resulting in the following.

TABLE 6. Sub-query and Facts retrieved

Query	Retrieved Facts
Who founded the Janatha Vimukthi Peramuna?	1. Dimuthu Bandara Abayakoon belongs to the Janatha Vimukthi Peramuna. 2. Rohana Wijeweera was the founding leader of the Janatha Vimukthi Peramuna. 3. Rohana Wijeweera led the Janatha Vimukthi Peramuna in an insurrection in 1971.

The LLM acts as a judge, analyzing the retrieved facts to determine that they are sufficient to answer the query, providing the final response as "Rohana Wijeweera".

However, the system can erroneously generate inappropriate sub-queries leading to an unending loop when relevant facts are absent from the Knowledge Web. Thus, our system allows an adjustable limitation on depth and degree of the recursion tree as shown in Figure 7.

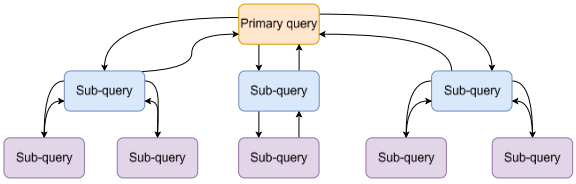

Fig. 7. Recursion Tree of the Fact Finding Agent. Here, the total depth of the Tree is 3 and the degree of the tree is 3.

IV. Experimental Setup

A. *Dataset*

With the objective of evaluating the validity of our system, a small Q&A dataset, derived from the MuSiQue Dataset [7] is used. As shown in Table 7, a sample of 70 answerable queries were chosen at random, with only the supporting passages to the answer being stored in the Knowledge Web.

TABLE 7. Contents of the Dataset

Questions	Passages	Keyword nodes	Fact nodes
70	140	1369	1426

B. *Implementation details*

The LLM being used is OpenAI's gpt-4o-2024-05-13 with temperature parameter set to 0; Neo4J Graph database and

Pinecone Vector database are used for the implementation of the Knowledge Web. OpenAI's text-embedding-ada-002 model is used for generating vector embeddings.

The threshold for the Keyword Resolution score is 0.8. For every query, top 5 facts are retrieved according to the Fact Ranking score.

C. Metrics

We report the Approximate Match (AM) of answers. An approximate match of answers is manually determined by ensuring that the generated answer alludes to the true answer mentioned within the dataset.

Consider Table 8, Paulsdale is a house museum where Alice Paul was born, located in Mount Laurel Township. Thus, although the location of Alice Paul's birth is mentioned as Mount Laurel Township in the dataset, Paulsdale being more specific is an acceptable answer.

TABLE 8. An example for approximate match

Query	Answer in Dataset	Generated answer
Where was Alice Paul born?	Mount Laurel Township	Paulsdale

D. Result

As a baseline, we have included the Approximate Match results obtained using the same dataset with Vector and Multi-query retrieval methods provided as features in LangChain, a suite of products to manage LLMs [8]. A generous restriction with chunk-size of 200 tokens, around 20 times the length of an average-decomposed fact in the Knowledge Web, simulates the restriction in the context length of LLM. Top – 5 retrieved documents and facts are considered to generate a final response.

As shown in Table 9, Knowledge Web with the Fact Finding Agent achieved an accuracy of 14.3% more than the multi-query and 17.2% more than the vector retrieval method.

TABLE 9. Results observed

RAG Technique (Top - 5)	AM
Vector Retrieval (Chunk size-200)	48.5%
Multi-query (Chunk size-200)	51.4%
Knowledge Web with Fact Finding Agent	65.7%

V. Discussion

A. Improving matching of keywords

Since the fact retrieval method includes a combination of vector similarity and Keyword Resolution Score to identify keywords within the Knowledge graph, a lack of indexed keywords within a query can pose failure. A possible solution would be to incorporate an LLM based synonym generation component to generate synonyms for the indexed keywords. Alternatively, the creation of a two-level vector index consisting of rich and simple keyword definitions can be leveraged to conditionally bypass the Keyword Resolution Score to ensure improved retrieval of facts.

B. Variance in structure of generated Knowledge Web

Although the structure of the Knowledge Web is anchored by the ordered pairs consisting of keywords and facts, the creation of the Knowledge Web is heavily influenced by prompts and the examples provided in them. Facts, keywords and the relationships within the Knowledge Web are processed from the uploaded documents using LLMs with the single shot learning principle. Single shot learning was preferred to obtain consistent outputs as well as reduce token usage to optimize cost. Our observations using GPT-4o as the LLM revealed that the samples included within the Knowledge Web generation must ideally be similar in structure to the unseen documents uploaded and their expected output of keywords, facts and relationships. Some document attributes considered to maintain consistent Knowledge Web generation are:

$$\text{Facts to document ratio} = \frac{\text{number of facts}}{document\ length}$$

$$\text{Keywords to document ratio} = \frac{\text{number of keywords}}{document\ length}$$

VI. Conclusion

Our Fact-Centric Knowledge Web offers a robust solution bringing the benefits of Knowledge Graph, while mitigating challenges faced by traditional Knowledge Graphs such as domain specificity and information loss. As Large Language Models continue to evolve, our approach provides a scalable and adaptable framework that can keep pace with these advancements, ensuring the continued reliability and effectiveness of RAG systems. Experimental results using a subset of the MuSiQue dataset demonstrate the potential efficacy of our system, achieving improvement in answer accuracy compared to the traditional vector and multi-query retrieval methods. However, further research and development are required to refine the system, address potential limitations, and compare its performance across larger and more diverse benchmarking datasets.

VII. References

[1] Lewis, P., Perez, E., & Klarquest, I. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, 945-955.

[2] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, Francesco Osborne, "Knowledge Graphs: Opportunities and Challenges", arXiv:2303.13948

[3] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, Percy Liang, Lost in the Middle: How Language Models Use Long Contexts, Transactions of the Association for Computational Linguistics (TACL), 2023

[4] Wang, X., Wei, J., Schuurmans, D., Le, Q. V., & Chi, E. H. (2022). "Large Language Models are Self-Consistent Reasoners". *arXiv preprint arXiv:2207.01780*.

[5] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). "Language models are few-shot learners" *Advances in Neural Information Processing Systems, 33*, 1877-1901.

[6] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, Ashish Sabharwal, "Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions", arXiv:2212.10509, ACL'23 Camera Ready

[7] Khot, T., Sabharwal, A., & Clark, P. (2022). "Musique: Multihop Questions via Single-hop Question Composition". *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2022)*, 2231-2243.

[8] LangChain: Available: https://github.com/hwchase17/langchain