Red Hen Lab GSoC 2024 Proposal
Knowledge Graph-Augmented RAG Systems for Improved Frame Blending Generation

Personal Information

Name: Kwan Kin, Chan

Email : chankwankin3@gmail.com

Github: https://github.com/kenchanLOL

Time Zone: CST(UTC -6)

Background

I possess two research experiences related to Large Language Models (LLMs), where I engaged deeply in literature reviews and hands-on implementation of LLM. Additionally, my academic background has provided me with a solid foundation in statistics.

- Education：
  Master of Computer Science, Texas A&M University (2023-2025)
  Bachelor of Engineer (Computer Science), The University of Hong Kong (2019-2023)
- Work Experience :
  1. Research Assistant, Intelligent Geometric and Visual Computing (IGVC) Lab @ TAMU
     - Develop Video Question Answering system for lecture videos using RAG and Knowledge Graph based Multimodal Model
     - Design experiments, measuring metrics and statistical analysis on evaluating Multimodal Language Models’ capability on video understanding and knowledge extraction
     - Explore VLM with ReAct prompting and CoT for Knowledge Graph Queries Generation

2. AI Engineer Intern, Hong Kong Hospital Authority AI Lab
   - Developed RAG-based medical Gradio Chatbot
   - Explore prompting techniques for different open-source LLMs (falcon, vicuna, llama2) on RAG conversation
   - Flagged an quantization implementation issue for huggingface's LLaMa2 with root cause analysis

Preparation

- Understanding on related technologies
  1. FrameNet :
     It is a database that map meaning to form based on the semantic theory. It consists of frames that include a set of elements and predicates, elucidating the relationships among these elements. Frame elements (FEs) are divided into core, which are frame-specific, and non-core, denoting general properties. Furthermore, the database encapsulates relationships between frames, such as parent-child linkages and inheritance patterns. This data structure seamlessly aligns with the knowledge graph (KG) paradigm, a structured format for storing knowledge, conceptual entities, and their interrelations.

     By harnessing research and technologies tailored for knowledge graph-enhanced Large Language Models (LLMs), we can adeptly retrieve and manipulate information for frame blending. This methodology not only positions frame blending as a prominent research topic within the LLM domain but also enhances explainability and controllability in the frame blending generation process. This is achieved by evaluating intermediate stages, such as the generation quality of FrameNet KG and KG queries for information retrieval.

2. LLM :

I aim to analyze the strengths and weaknesses of open-source Large Language Models (LLMs) from two angles: application-wise and implementation-wise. In terms of applications, I have conducted tests to evaluate the LLMs' proficiency in processing FrameNet data and constructing knowledge graphs. The detailed results and methodologies of these evaluations are documented in a GitHub repository. For implementation, my focus has been on comparing the models' sizes and the range of supported acceleration packages.

1. Application

This experiment was carried out on Chatbot Arena platform supported by LMSYS. Given the context length limitation, the FrameNet data was inputted as raw text instead of in XML format. To ensure fairness, each model produced 3 output samples with the evaluation based on the best one. Results from GPT-4 and Claude3 were also assessed to serve as reference benchmarks. The prompts utilized were based on the ReAct + CoT [1] and Alpaca[2] :

{FrameNet text}

### Instrustion

Given the above information, generate examples of graph queries to build the following relationships in knowledge graph with Neo4j syntax :frame and element node, Link element to frame, Inter-element Relationship, Frame Relations and Hierarchies, Metaphorical and Contrastive Relationships

### Output Format

Thought : <thought>

Action : <queries example>

In summary, the majority of models exhibit instability in generating high-quality and accurate knowledge graph (KG) queries. Observations indicate that models with larger sizes, exceeding 100 billion parameters, tend to produce better-quality outputs in general. Conversely, among the smaller models, those with around 7 billion parameters, Mistral significantly outperforms Gemma, showcasing a robust ability to handle inter-element relationships, which are both challenging and critical.

Model	Remarks
GPT-4	• Generate queries for all types of relationships with correct entity name and relationship. Able to extract keyword to describe inter-element relationships
Claude-3	• Fail to generate queries that accurately describe inter-element relationships and Metaphorical and Contrastive Relationships.
Mixtral-8x7b	• Most similar overall queries quality as GPT-4
Mistral-7b	• Most similar queries quality on inter-element relationships as GPT-4
LLaMa2-70b	• Fail to generate correct queries for Frame Relations and Hierarchies
Gemma-7b	• Fail to generate correct queries for most relationships except Link element to frame
Dbrx-instruct	• Second most similar overall queries quality as GPT-4, less stable compared to Mixtral-8x7b

2.Implementation

Model	Remarks
Mixtral-8x7b	• Using Sparse Mixture of Expert architecture allows it to leverage up to 45B parameters but only uses about 12B during inference (active parameter)
Mistral-7b	• A downgrade version of Mixtral-8x7b
LLaMa2	• Wide range of model size available (2B, 7B, 70B) • Inference acceleration is supported by the Llama.cpp library.
Gemma	• Available for 2 model sizes (2B, 7B) • Largest token vocabulary size lead to around 10% saving on tokens for same length of context
Dbrx-instruct	• Largest model size (132B parameter with 36B active parameter)

In conclusion, Mistral-7b emerges as the most promising option, thanks to its smaller size, which facilitates easier fine-tuning, and its strong performance on targeted tasks. If more computational resources become available, we could also consider Mistral-8x7b and Dbrx. However, Parameter Efficient Fine-tuning (PEFT) may be necessary for fine-tuning these larger models.

Proposal

Goal

1. Construct a comprehensive FrameNet knowledge graph:
   While previous research has primarily built FrameNet-based graphs for applications like Named Entity Recognition[3], focusing mainly on frame and frame element (FE) relations[4], our project aims to delve deeper. I plan to extract more complex relationships, such as inter-element relationships, frame relations and hierarchies, and metaphorical and contrastive relationships. These aspects have been overlooked due to their difficulty. By employing Large Language Models (LLMs), we can simplify this process, facilitating the development of a more comprehensive knowledge graph for FrameNet.

2. Develop frame blending with a KG-enhanced RAG system :
   I am keen on advancing the Retrieval-Augmented Generation (RAG) system by integrating various technologies like knowledge graphs and LLM agent systems for the frame blending task. This enhanced system would enable LLMs to perform graph queries, iteratively fetching relevant information for frame blending, such as identifying the common parent of two frames. This approach would render the generation process more explainable and controllable, while also minimizing the occurrence of hallucinations due to insufficient information.

3. Developing chatbot application for frame blending (depends) :
   Utilize the instructional capabilities of LLMs to enable users to input instruction in plain text, enabling interactive generation or refinement of frame blending outcomes.

Plan

1. Enhance understanding of KG-enhanced RAG, focusing on Neo4j and its compatibility with Langchain, through courses offered by DeepLearning.AI.

2. Get familiar with FrameNet by exploring prompt engineering for stable generation and develop a pipeline (KG Constructor) to convert XML data into a Knowledge Graph

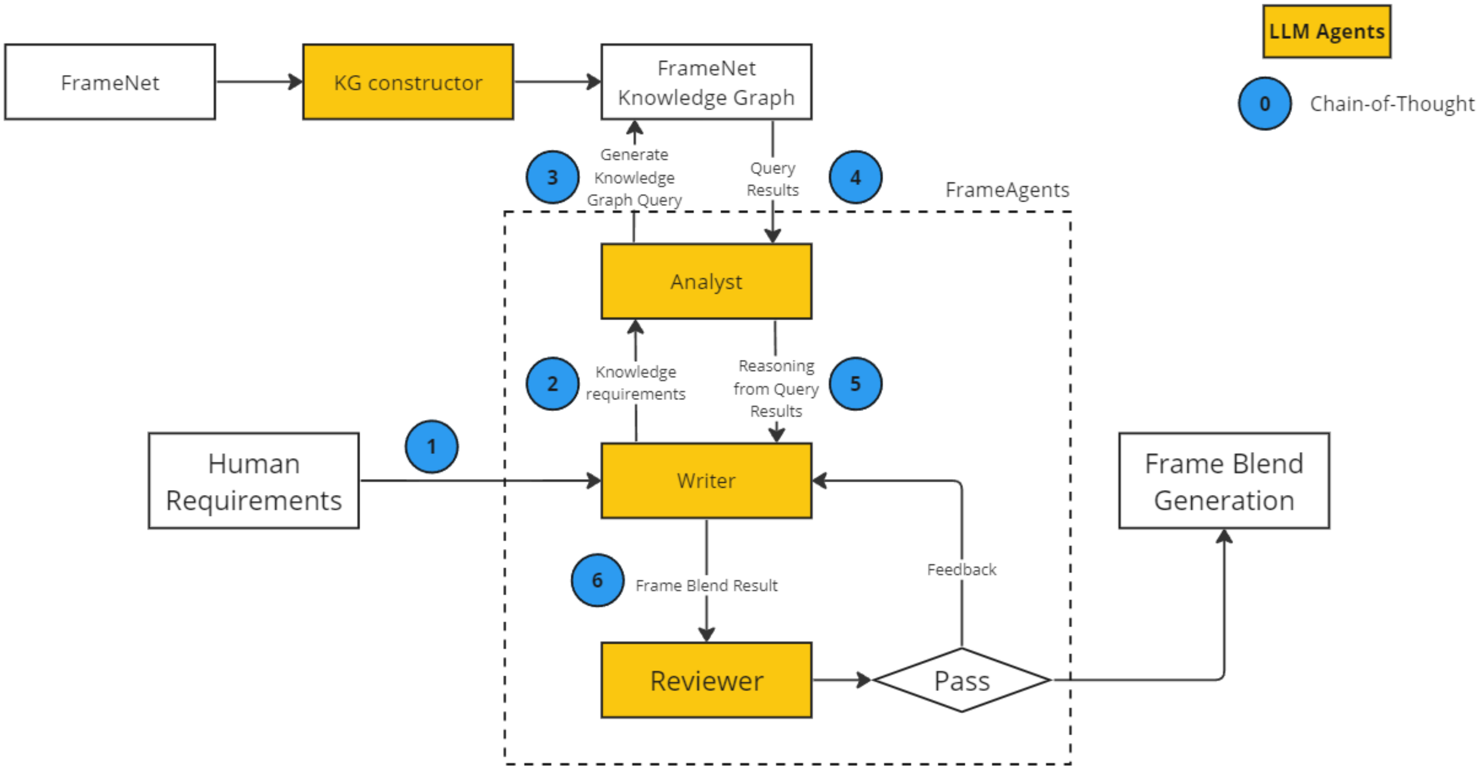
3. Design and create a Gradio application that features visualization and customization options to facilitate later experimental processes.

4. Build the LLM Frame Blending system :

   The recent advancements in Chain of Thought (CoT) processing, particularly within the code generation domain, indicate that the self-refinement CoT approach and few-shot prompting are instrumental in improving the quality of outputs. This methodology prompts the model to produce outputs while simultaneously generating feedback. However, utilizing the same model for both output and feedback generation might degrade its performance over time. To address this, the concept of a Self-Collaboration model has been introduced, which segregates the roles within the LLMs, each provided with distinct prompts, materials, and few-shot examples [5].|

   Building on this concept, I propose a similar method for constructing an LLM Frame Blending system. By deconstructing the complex 'Chain of Thought' into segments for frame blending results, three distinct agents are designed : Analyst, Writer, and Reviewer. These agents engage in iterative interactions, particularly between the Analyst and Writer, and the Writer and Reviewer.

The Writer's role is to synthesize knowledge descriptions deemed relevant based on the human requirements (step 1) and knowledge requirements (step 2), and subsequently create frame blend examples utilizing the reasoning from query results (step 5), which are then assessed by the Reviewer (step 6). The Analyst generates queries based on the initial knowledge description and human requirements (step 3) and parses the query results into a report that informs the Writer's creation of frame blend examples (step 4). The Reviewer evaluates whether the examples from the Writer meet the quality standards set by human operators and provides feedback if they fall short.



Overview of the Frame Blending system

Timeline

1st May	Proposal result release	• Finish exams and assignments until 5th May • Finish the DeepLearning.AI course
27th May	Start of Program	• Finish literature review, first prototype of KG construction pipeline
3rd June	Milestone #1	• Goal 1 1. Setup Environment for LLM inference 2. Systematic Evaluation of LLM models 3. Systematic Evaluation of KG Generation Prompt
10th June	Milestone #2	• Goal 1 1. Complete the KG generation pipeline • Goal 2 1. Explore the embedding model for RAG system
24th June	Milestone #3	• Goal 2 1. Design Evaluation Metric 2. Implement the Analyst agent
1st July	Milestone #4	• Goal 2 1. Implement the Writer agent 2. Provide manual feedback to test the performance on self-refinement framework
8th July	Midterm evaluation	• Summarize previous work as blog post and report
22nd July	Milestone #5	• Goal 2 1. Implement the Reviewer agent 2. Test the whole sytem

29th July	Milestone #6	• Goal 2 1. Build a Gradio Interface for general user
5th August	Milestone #7	• Goal 3 1. Experiment on interactive chatbot of frame blending generation
12th August	Final Week	• Documenting the project including code comments, API usage, System Design, Deployment Process and User Guides
19th August	Work Report	• Summarize all the work as blog post and report
26th August	End of Program	• Mentors submit final GSoC contributor evaluations

Reference List :

1. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Chen, Y. (2022b). REACT: Synergizing reasoning and acting in language models. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2210.03629
2. Stanford CRFM. (2023, March 13). https://crfm.stanford.edu/2023/03/13/alpaca.html
3. Cécile Robin, Atharva Kulkarni, and Paul Buitelaar. 2023. Identifying FrameNet Lexical Semantic Structures for Knowledge Graph Extraction from Financial Customer Interactions. In Proceedings of the 12th Global Wordnet Conference, pages 91–100, University of the Basque Country, Donostia - San Sebastian, Basque Country. Global Wordnet Association.
4. Zheng, C., Chen, X., Xu, R., & Chang, B. (2022). A Double-Graph based framework for frame semantic parsing. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. https://doi.org/10.18653/v1/2022.naacl-main.368
5. Dong, H., Bu, Q., Zhang, J. M., Lück, M., & Cui, H. (2023). AgentCoder: Multi-Agent-based Code Generation with Iterative Testing and Optimisation. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2312.13010