

GraphAM: Graph Database-Integrated Active Memory for Generative Language Models

2023.08

Donggyu Lee

ldg@takeup.cc

Hankuk Academy of Foreign Studies

synapse, take up

Abstract

This study presents an active memory algorithm that generates responses in generative language models using graph databases. The development of generative language models has picked up pace recently, and there are many commercial services available. However, generative language models are limited by problems such as hallucination, low accuracy and reliability, and limitations in contextualizing and remembering. It is expensive and requires a lot of resources to develop pre-training datasets or fine-tune the base model to address these problems. Instead, well-designed prompts can be used to achieve the desired response, but this requires prompt engineers or training, as well as a thorough understanding of generative language models.

All conversations are saved in a graph database to build a memory, and when a user asks a question, it proactively identifies the information it needs and pulls it and its neighbors from the graph database for reference as it generates an answer to the question. This approach streamlines the generation of natural language that disentangles complex and interconnected information in the real world. Research has shown that answering questions based on real-world information increases the efficiency and usability of generative language models in processing information and generating answers.

In addition, the memory assist algorithm of the graph database converts various text datasets, not only conversations, into property graph models that can be updated in real time, and provides diverse and accurate information to the generative language model, enabling it to generate accurate responses through diverse information while reducing the size of the language model, thereby increasing efficiency and speed.

Table of contents

I. Theoretical Background

A. Generative Language Model

B. Model Tuning

C. Database

II. Methods

A. Environment

B. Algorithm

III. Results

A. Learning

B. Answering

IV. Discussion

I. Theoretical Background

A. Generative Language Model

Generative Language Models are a type of artificial intelligence model that can generate natural language. These models are trained on large datasets of text, which allows them to learn the patterns and structures of natural language.

GPT

GPT (Generative Pre-trained Transformer) is a series of generative language models developed by OpenAI. These models are based on the transformer, which is a type of machine learning model developed by the Google Brain team. The GPT models are pre-trained on vast private datasets of text, such as Wikipedia, and then fine-tuned for specific tasks, including question answering or text completion. The most recent model, GPT-4, was released in 2023 and is now publicly available via the OpenAI API.

Compared to GPT-3, which was previously a high-performing generative language model based on zero-shot transformer technology, includes stricter guidelines for learning and answering questions. In addition to Reinforcement Learning from Human Feedback, which was fine-tuned in GPT-3.5 for generating answers to human questions, it is built to follow more rules and stricter guidelines by further censoring the user's prompt input and the language model's answers.

B. Model Tuning

Pre-training

The creation of a language model based on training with a large data set of text. This process requires the accumulation of the entire large dataset, and often involves labeling and cleaning to refine the data. Due to the use of large data sets, developing a model specialized for specific tasks or instructions is difficult, and it takes a lot of time and resources to retrain the model if the data set changes. This creates challenges for not only adding new information in real time but also on a regularly updated basis.

Fine-tuning

It is the process of additional training of a pre-trained model to improve its accuracy or efficiency on a specific task. This requires a large data set for the specific task to be fine-tuned, which requires intensive time and resources.

Therefore, while it might be a lighter task than building a model with pre-training, it still requires a lot of time and resources, and in some cases it is more difficult because it requires precise tuning for

a specific task. As a result, fine-tuning cannot memorize conversations or context that the user creates in real-time.

Prompt Engineering

This method does not directly affect the model through training as in the previous methods. Instead, it controls the information provided and the method of questioning and input through prompts and sentence structure. This necessitates efficient organization of information within limited tokens, minimizing unnecessary questions or information and requiring precise input that matches intent. Through this, the accuracy of responses can be improved while reducing the cost and time wasted on inaccurate results.

C. Database

Graph Database

A graph database stores and processes data in a property graph model consisting of nodes and edges. This structure is ideal for handling complex and close interrelationships in various fields, including social media, networks, and geographic information systems. The efficient retrieval of information surrounding world through node relationships makes it a valuable.

II. Methods

A. Environment

Language

This research was implemented using Javascript. Generative language models like GPT-4, which have 1 trillion parameters and train on massive language datasets, take a long time to process prompts and output results. This involves multiple processing steps, such as reading and writing information, and requires a language capable of asynchronous processing. Through asynchronous processing, the main thread can continue to operate without being interrupted while handling I/O operations.

The study was developed using Node.js. Node.js is a Javascript runtime based on Chrome V8 and forms the foundation for most Javascript frameworks.

DB

Neo4j is the most commonly used graph database, exhibiting excellent performance in querying complex relationships based on its high performance graph database capabilities. As an open-source project, it provides a large community and resources, as well as various third-party drivers and languages, allowing the use of diverse clients or languages.

By using the Cypher query language, it is possible to query, analyze or store property graph data. It shows superior performance in queries that require JOIN operations compared to traditional RDBMSs. It has flexibility with a flexible schema that can store data in different forms.

B. Algorithm

Understand

Input is prompt that user write for question. You need to extract or create keyword, label, relationship or properties and make cypher query that will search database to assist ai for answering. Must not create new data in this step. To avoid repetition and to group the relationships and related nodes under separate keys. If you can't assist of question, just answer QUIT without explanation. You can answer only in cypher query language.

System prompt

This is the stage of understanding the question input by the user. The user's question is analyzed actively to determine necessary information and specific elements for answering, extracting keywords and properties from the graph database. An query can be written actively to find nodes and relationships that match, include, or are associated with these keywords, properties, or labels.

In this study, we use OpenAI's generative language model GPT-4 to generate Neo4j's query language - Cypher Query Language - using the prompts provided above.

Additionally, this prompt is controlled to only output the Cypher Query Language in response to the user's question, and if an answer is not possible, it returns only a designated separate word to prevent any impact on the next function call. Also, because the return value of a query includes not only the requested information but also surrounding information, we use a separate key to group different relationships and nodes to avoid the same information from being returned multiple times.

Think

The query returned through the Understand stage is input into the graph database to acquire information for answering the question. At this point, the data returned from the database is formatted as a list of key-value pairs to efficiently reference when the generative language model generates an answer to the user's question.

Answer

answer user message refer prompt & memories/Pre written prompt=\${preprompt}/Related memories from chat history=\${think}
System prompt

The user's question and the information obtained in the Think stage for answering the question are input into the generative language model to generate an answer to the question. Since answers are generated based on additional information obtained in the Think stage, accurate answers that include this information can be made possible due to related data located within the input prompt.

This process can be conducted by selecting a generative language model that one wishes to receive an answer from. In this study, we used OpenAI's GPT-4 model to generate answers to questions through the prompt below.

Memorize

extract or create keyword, label, relationship or properties from question & answer which need to memorize for future answers. Each node has relationship with general keyword. Use MERGE to avoid duplicate data & do not use same variable for avoid error. If refer exist data, increase weight properties and update scarce data, but if new data, weight properties will be essential & data from question increase weight double. Answer in only cypher query language.	question=\${user_msg}/answer=\${answer}
System prompt	user input

The language model proactively extracts keywords, labels, relationships or properties that are deemed necessary for generating future answers, based on the user's question and the generative language model's answer, and stores them in the graph database. It generates a query that can store information in the graph database by inputting the user's question and the language model's answer. The generated query is input into the graph database for storage.

Beyond simple information storage, it updates importance or access frequency of information to use as an indicator for fetching and utilizing information through priority during future Understand stages. For example, if an answer is generated using already stored information, it increases the weight attribute of this information so it can be fetched first during future Understand stages and serves as an indicator for understanding importance of information when generating answers with a generative language model.

Furthermore, because it continuously creates and updates relationships with all nodes, utilization and value of existing information continues to rise. Even without directly fetching this specific piece of data, its utility increases as it can be fetched as surrounding information due to formed relationships.

III. Results

A. Learning

Active transfer text to property graph

Remember this text for future answers.	Generative Pre-trained Transformer 4 (GPT-4) is a multimodal large language model created by OpenAI, and the fourth in its series of GPT foundation models.[1] It was initially released on March 14, 2023, and has been made publicly available via the paid chatbot product ChatGPT Plus, and via OpenAI's API. As a transformer-based model, GPT-4 uses a paradigm where pre-training using both public data and data licensed from third-party providers is used to predict the next token. After this step, the model was then fine-tuned with reinforcement learning feedback from humans and AI for human alignment and policy compliance. Observers reported that the iteration of ChatGPT using GPT-4 was an improvement on the previous iteration based on GPT-3.5, with the caveat that GPT-4 retains some of the problems with earlier revisions. GPT-4 is also capable of taking images as input, though this feature has not been made available since launch. OpenAI has declined to reveal various technical details and statistics about GPT-4, such as the precise size of the model.
Pre prompt	user input

<pre>MATCH (m:Model {name: 'Generative Pre-trained Transformer 4', alias: 'GPT-4'})-[:CREATED_BY]->(o:Organization {name: 'OpenAI'}), (m)-[:RELEASE_DATE]->(d:Date {date: 'March 14, 2023'}), (m)-[:AVAILABLE_THROUGH]->(p:Product {name: 'ChatGPT Plus'}), (m)-[:USES]->(t:Technique {name: 'Transformer-based model'}), (m)-[:PRE_TRAINED_ON]->(pd:Data), (m)-[:FINE_TUNED_WITH]->(ft:Technique {name: 'Reinforcement learning'}), (m)-[:IMPROVEMENT_ON]->(prev:Model {name: 'GPT-3.5'}), (m)-[:HAS_ISSUE]->(i:Issue), (m)-[:CAPABLE_OF]->(f:Feature {name: 'Taking images as input'}) RETURN m, o, d, p, t, pd, ft, prev, i, f</pre>
Query for Think stage (generated @ Understand stage)

I have stored the information about Generative Pre-trained Transformer 4 (GPT-4). I understand that it is a multimodal large language model created by OpenAI, released on March 14, 2023. It's available via the paid chatbot product ChatGPT Plus, and OpenAI's API. It uses a transformer-based model, pre-training using both public data and data licensed from third-party providers to predict the next token. It was then fine-tuned with reinforcement learning feedback from humans and AI. It's an improvement on the previous iteration based on GPT-3.5, but still retains some of the problems with earlier revisions. GPT-4 can take images as input, but this feature has not been made available since launch. OpenAI has not revealed various technical details and statistics about GPT-4, such as the precise size of the model.

Answer (generated @ Answer stage)

```
MERGE (gpt4:Model {name: 'Generative Pre-trained Transformer 4 (GPT-4)'})
ON CREATE SET gpt4.created_by = 'OpenAI', gpt4.release_date = 'March 14, 2023',
gpt4.available_via = ['ChatGPT Plus', 'OpenAI API'], gpt4.model_type = 'transformer-based',
gpt4.pre_training_data = ['public data', 'data licensed from third-party providers'],
gpt4.fine_tuning = 'reinforcement learning feedback from humans and AI', gpt4.previous_version
= 'GPT-3.5', gpt4.image_input_capability = 'Yes', gpt4.image_input_availability = 'No',
gpt4.technical_details_revealed = 'No', gpt4.weight = 2
ON MATCH SET gpt4.weight = gpt4.weight + 2
```

```
MERGE (openai:Company {name: 'OpenAI'})
ON CREATE SET openai.weight = 1
ON MATCH SET openai.weight = openai.weight + 1
```

```
MERGE (chatgpt_plus:Product {name: 'ChatGPT Plus'})
ON CREATE SET chatgpt_plus.weight = 1
ON MATCH SET chatgpt_plus.weight = chatgpt_plus.weight + 1
```

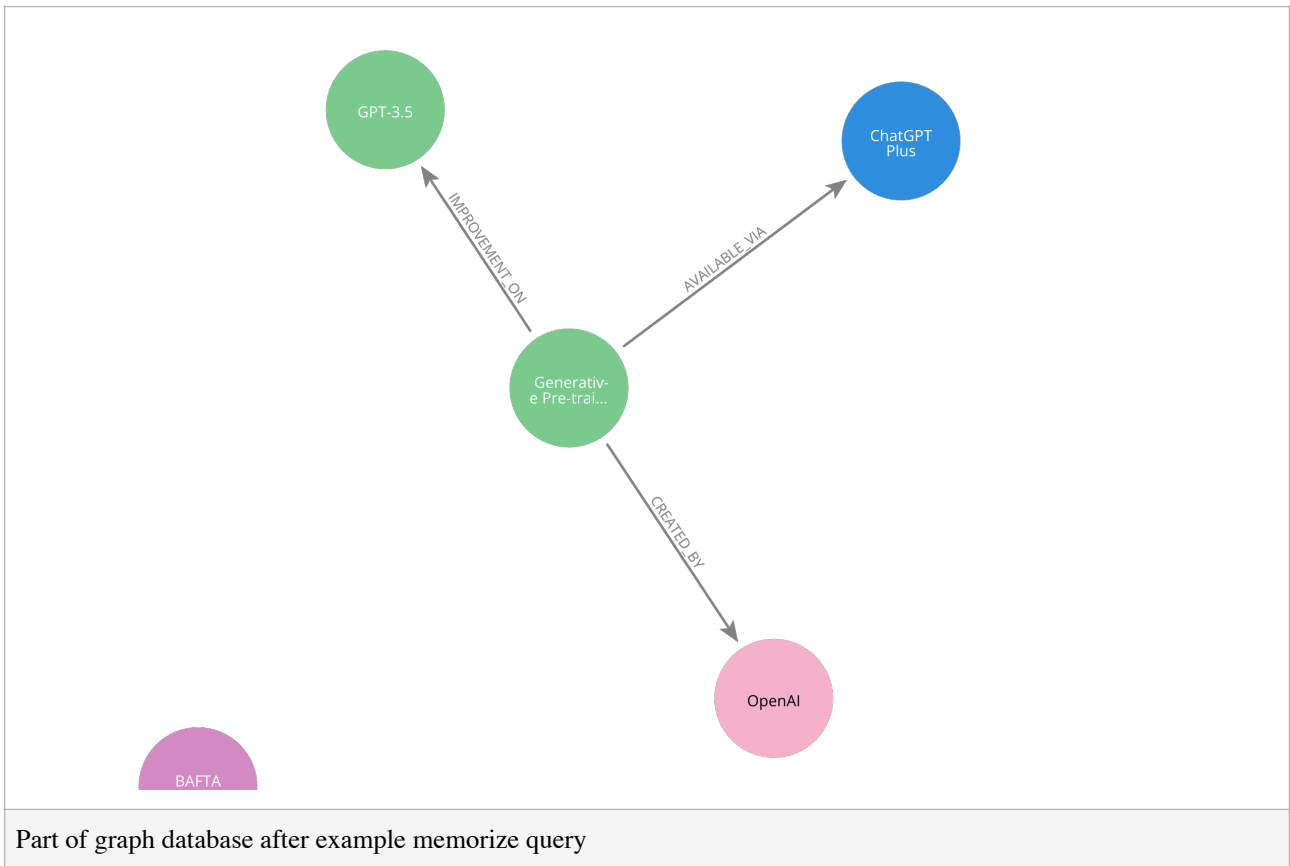
```
MERGE (gpt3_5:Model {name: 'GPT-3.5'})
ON CREATE SET gpt3_5.weight = 1
ON MATCH SET gpt3_5.weight = gpt3_5.weight + 1
```

```
MERGE (gpt4)-[:CREATED_BY]->(openai)
```

```
MERGE (gpt4)-[:AVAILABLE_VIA]->(chatgpt_plus)
```

```
MERGE (gpt4)-[:IMPROVEMENT_ON]->(gpt3_5)
```

Query @ Memorize stage



As described above, when information is given to the generative language model, it actively understands the content and user's intent and stores it in the graph database. Information delivered in natural language sentences is converted into nodes and relationships of many property graph. If this information already exists in the database, it creates relationships between existing nodes and new different pieces of information, recording that they are related. Additionally, nodes and relationships that are read or written repeatedly have their weight attribute values increased, identifying them as priority references for future information searches.

B. Answering

Answer compare

Do you know about gpt-4?

Question for answer compare

<p>As an AI developed by OpenAI, I'm based on the GPT-3 model. As of my last update, GPT-4 has not been released or detailed. However, it's expected that GPT-4 would be an even more advanced version of the current model, with improved capabilities in language understanding and generation. Please check OpenAI's official announcements for the most accurate and up-to-date information.</p>	<p>Yes, GPT-4 is a multimodal large language model created by OpenAI. It's the fourth in the series of GPT foundation models. It was released on March 14, 2023, and is available via the paid chatbot product ChatGPT Plus, and OpenAI's API. It uses a transformer-based model where pre-training is done using both public data and data licensed from third-party providers to predict the next token. The model is then fine-tuned with reinforcement learning feedback from humans and AI for human alignment and policy compliance.</p>
<p>Pure GPT-4 (temperature: 0)</p>	<p>GPT-4 with GraphAM (temperature: 0)</p>

Through the memory assistance algorithm using the graph database, it immediately finds the information needed according to the question and assists in generating answers for the generative language model. This research shows that if a graph database is used as a memory assistance algorithm for generative language models, it can update latest data in real time without any modifications to the generative language model. Additionally, by storing information as an property graph model through a graph database, it can fetch related information along with natural language sentences or of information similar to human thinking patterns, thereby providing actual data for answers.

IV. Discussion

For generative language models, it is difficult to include all the information and perform all the tasks you want. However, through the memory assistance algorithm using a graph database, it can improve accuracy and performance for specific tasks more simply than fine-tuning across all domains. Additionally, continually updating data can be persistently maintained in real-time with fewer resources. Generative language models focus only on generating language, while algorithms take over for specific tasks or information, leading to higher performance of efficient language models with fewer parameters.

Through the memory assistance algorithm of this study, the generative language model can receive related information from all past conversation data relevant to the current question, enabling more personal-friendly or purpose-driven answers. Also, it can reduce hallucination problems of generative language models since it can reference not only past conversations but also other data stored by property graph.

The current study only includes content about storing and fetching conversations with a language model as an property graph model. However, if various natural language text data are converted into property graph containing relationship and property information and connected to generative language models through this study's graph database memory assistance algorithm, we believe that it could increase accuracy of information by linking existing diverse information with artificial intelligence and decrease hallucinations.

V. References

[1] OpenAI, “GPT-4 Technical Report”, Mar, 2023.

[2] Ashish Vaswani and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N. Gomez and Lukasz Kaiser and Illia Polosukhin, “Attention Is All You Need”, Jun, 2017.

[3] Long Ouyang and Jeff Wu and Xu Jiang and Diogo Almeida and Carroll L. Wainwright and Pamela Mishkin and Chong Zhang and Sandhini Agarwal and Katarina Slama and Alex Ray and John Schulman and Jacob Hilton and Fraser Kelton and Luke Miller and Maddie Simens and Amanda Askell and Peter Welinder and Paul Christiano and Jan Leike and Ryan Lowe, “Training language models to follow instructions with human feedback”, Mar 2022.