Large Language Models and Data Streams

Seminar *Data Stream Management and Analysis*

July 15, 2024

Silyu Li

Juniorprofessur für Datenstrom-Management und -Analyse

RWTH AACHEN UNIVERSITY

Introduction

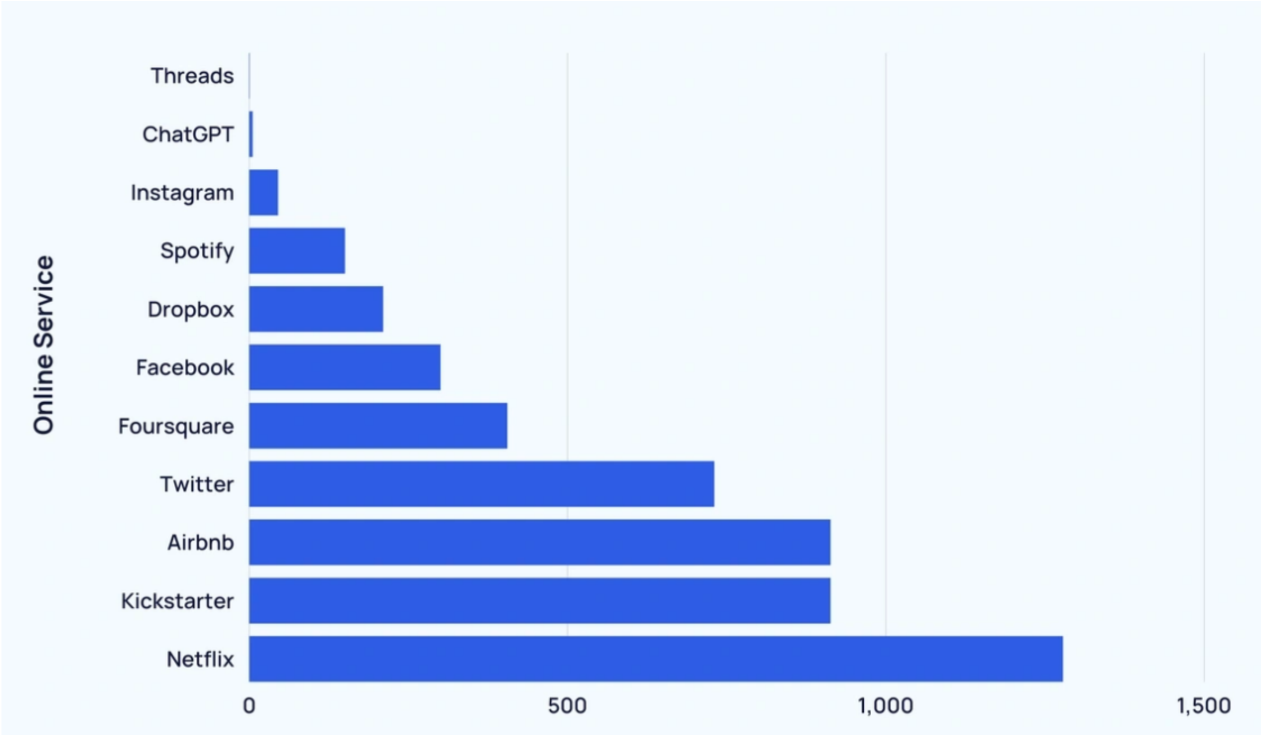
Background

Relevant concepts of AI models and technologies[a]

- LLM and AI have become very prominent topics in recent years.

[a]Source: https://www.cortical.io/blog/chatgpt-and-large-language-models-the-holy-grail-of-enterprise-ai/

Background



Time needed for different Apps to have 1 million users[1]

[1] Source: https://explodingtopics.com/blog/chatgpt-users

Juniorprofessur für Datenstrom-Management und -Analyse
RWTH AACHEN UNIVERSITY

Introduction

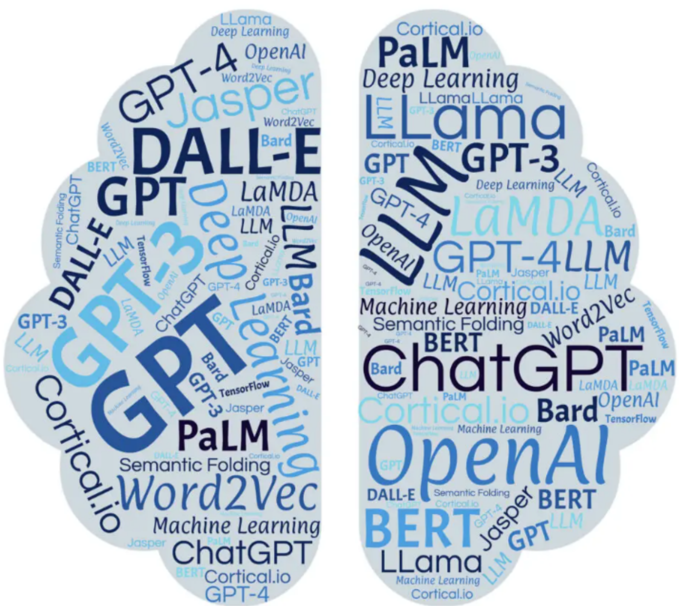
Background

Relevant concepts of AI models and technologies[a]

- LLM and AI have become very prominent topics in recent years.
- Wide usage and significant competence in QA, content generation, translation, text classification and other tasks [1].

[a]Source: https://www.cortical.io/blog/chatgpt-and-large-language-models-the-holy-grail-of-enterprise-ai/

Background



Relevant concepts of AI models and technologies[a]

- LLM and AI have become very prominent topics in recent years.
- Wide usage and significant competence in QA, content generation, translation, text classification and other tasks [1].
- Most models are ”static” [2].

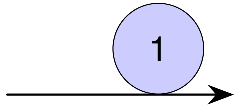
[a]Source: https://www.cortical.io/blog/chatgpt-and-large-language-models-the-holy-grail-of-enterprise-ai/

Content

- Evolution of language models and the techniques behind them
- Training process of several LLMs
- Need, benefits, challenges and use cases of combining LLM with data streams

Large Language Models

Simple Models



- N-Gram [3]
- Calculate the next word's probability based on the N-1 previous sub-words.
- Poor performance on large documents and large N.

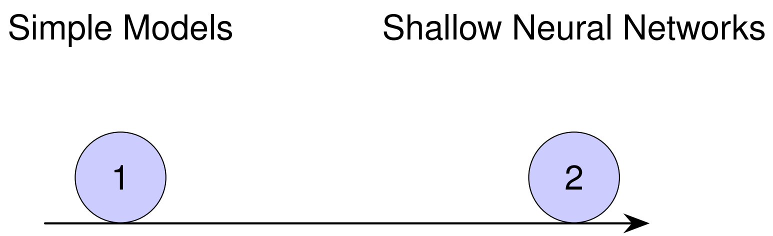
N-Gram

Example: Tri-Gram

Based on a text corpus, predict the next word of ”I read”

$$P('' I'', '' read'', '' x'') = Count('' I'', '' read'', '' x'')/Count('' I'', '' read'')$$

Large Language Models



- Word2Vec [4]
- Map words into vector space.
- Similar words have a closer distance.
- Can capture contextual information.
- Better performance on large documents.

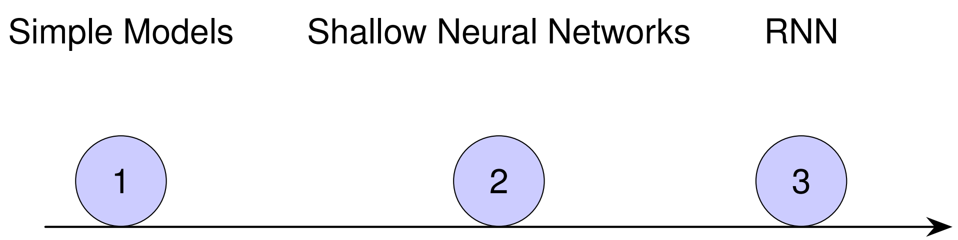
Word2Vec

Example

"I read a" $\Rightarrow V_I + V_{read} + V_a$

Compare similarity between $V_I + V_{read} + V_a$ and $V_I + V_{read} + V_a + V_x$

Large Language Models



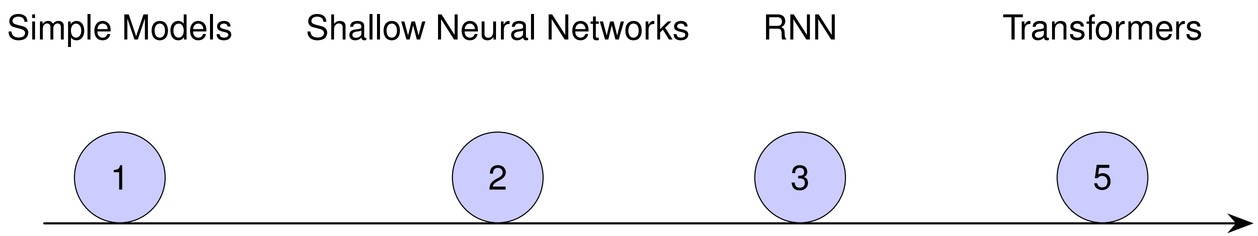
- Seq2Seq [5]
- Specially designed architecture to obtain long-term dependencies.
- Much better performance on large documents, can work with different input lengths.
- But still problematic on very large documents.

Seq2Seq

Example

"I read a book . . . " $\Rightarrow \{''\textit{I}''\} \Rightarrow \{''\textit{read}''\} \Rightarrow \{''\textit{a}''\} \Rightarrow \{''\textit{book}''\} \Rightarrow \ldots$

Large Language Models



- Transformer [6]
- Can process the entire input sentence simultaneously.
- Very good performance on large documents, more efficient and powerful.

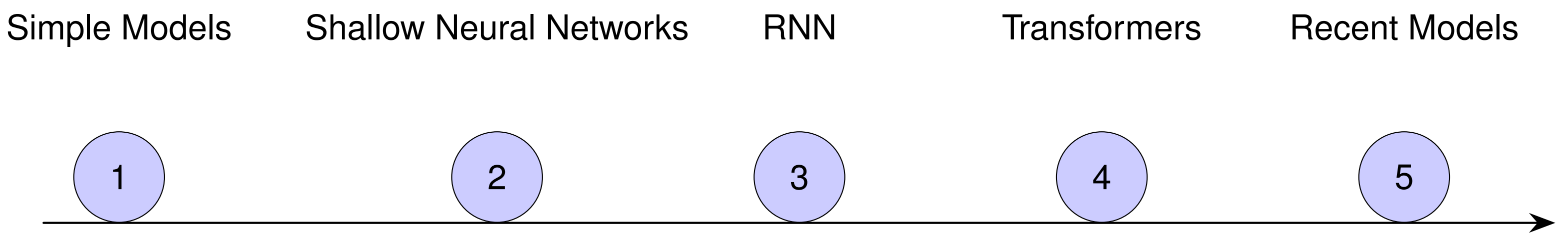
Transformer

Example

"I read a book . . . " $\Rightarrow \{''\, I_1''\,''\, read_2''\,''\, a_3''\,''\, book_4''\ \ldots\}$

Introduction

Large Language Models



- BERT, ChatGPT, LLaMA. . .
- Based on the transformer architecture.
- Trained on massive datasets and fine-tuned with downstream tasks.

Training Process of LLMs

Tokenization

- Purpose: Segmenting input text into tokens, which are elements from the vocabulary that the model knows.
- BERT: ”I read a book” $\Rightarrow$ [”I”, ”re”, ”##ad”, ”a”, ”book”] [8]
- GPT2, LLaMA: ”I read a book” $\Rightarrow$ [. . . , ”re”,”ad”,. . .] $\Rightarrow$ [. . . , ”read”,. . .] [9]

Training Process of LLMs

Pre-training

Given an unlabeled corpus of tokens $U = \{u_1, ..., u_n\}$ as a training dataset, the core idea of the pre-training phase is to predict the next token u_i for a sequence $\{u_{i-k}, ..., u_{i-1}\}$, specifically by maximizing the likelihood:

$$L_1(U) = \sum_i \log P(u_i | u_{i-k}, ..., u_{i-1}; \Theta)[10] \tag{1}$$

k refers to the context window size and Θ is the parameter of the neural network with which the conditional probability P is modeled.

Example: BERT [7]

- Next Sentence Prediction (Pre-training phase)
  - BERT is trained to predict if the second sentence B (of longer sentence A, B) is real, with B being replaced by a random sentence half the time.
  - ”I read a book”, ”it is amazing” or ”I read a book”, ”The weather is bad”.

Fine-tuning

Given a labeled dataset C, where each instance of C has a sequence of tokens $\{c^1, \ldots, c^m\}$ and a label y, the goal of the fine-tuning phase is to maximize the following likelihood:

$$L_2(U) = \sum_{(c,y)} \log P(y|c^1, \ldots, c^m)[10] \tag{2}$$

Example: BERT [7]

- Next Sentence Prediction (Pre-training phase)
  - BERT is trained to predict if the second sentence B (of longer sentence A, B) is real, with B being replaced by a random sentence half the time.
  - ”I read a book”, ”it is amazing” or ”I read a book”, ”The weather is bad”.
- Question Answering (Fine-tuning phase)
  - Bert uses the Stanford Question Answering Dataset that contains 100k question/answer pairs.

A Comparison of Different Models

Model Name	BERT	GPT-2	LLaMA	GPT-3,5/ 4
Developer	GoogleAI	OpenAI	MetaAI	OpenAI
Release Date	2018	2019	2023	2022/2023
Nr. of Parameters	110 M/ 340 M	1,5 B	7-65 B	175 B/1,7 T
Training Data	Wikipedia(en) & BookCorpus[2]	WebText	Various open-source datasets	WebText
Open-sourced	Yes	No	Yes	No
Major Applications	QA	Text generation & Translation	Text generation & QA & Translation	Content generation & QA & Translation

Comparison of Large Language Models

[2] https://en.wikipedia.org/wiki/BookCorpus

A Comparison of Different Models

Model Name	BERT	GPT-2	LLaMA	GPT-3,5/ 4
Developer	GoogleAI	OpenAI	MetaAI	OpenAI
Release Date	2018	2019	2023	2022/2023
Nr. of Parameters	110 M/ 340 M	1,5 B	7-65 B	175 B/1,7 T
Training Data	Wikipedia(en) & BookCorpus[3]	WebText	Various open-source datasets	WebText
Open-sourced	Yes	No	Yes	No
Major Applications	QA	Text generation & Translation	Text generation & QA & Translation	Content generation & QA & Translation

Comparison of Large Language Models

The performance of a model increases exponentially when either one of the following factors are scaled up: the number of model parameters, the size of training datasets and the available computational resource [18].

[3] https://en.wikipedia.org/wiki/BookCorpus

Data Stream

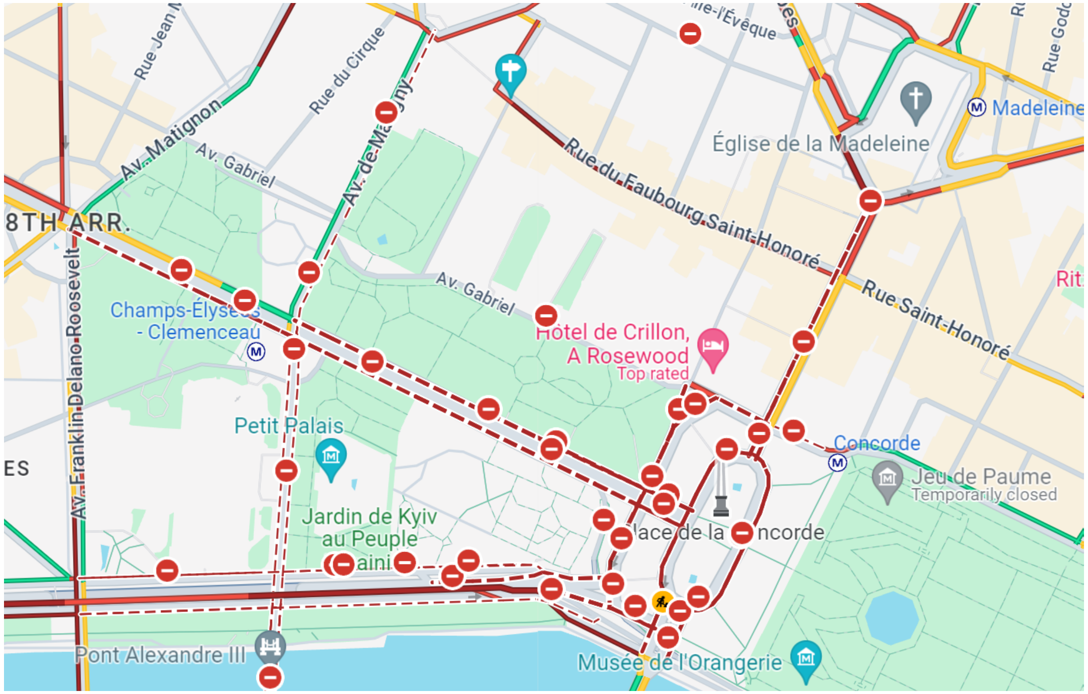
Definition 1

A data stream S is an unbounded, potentially infinite multiset of data stream elements (s, τ), where $\tau \in \mathbb{T}$. $\mathbb{T}$ is a timestamp attribute with values from a monotonic, infinite time domain $\mathbb{T}$ with discrete time units. [11]

Limitations of Pre-trained LLMs

- Lack of knowledge beyond the scope of their training datasets.
- The performance is likely to gradually degrade over time when finetuned on new tasks [12].
- Extremely computationally expensive to be re-trained.

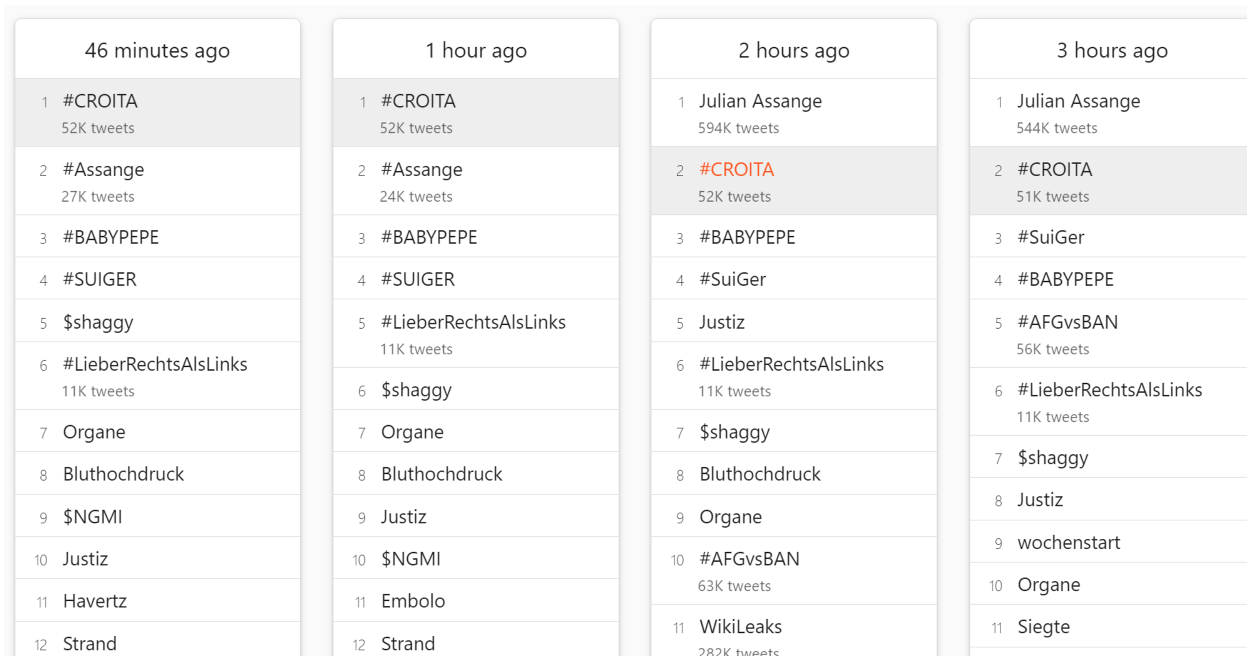
Application Fields: Traffic Data



A screenshot of traffic in Paris using Google Maps[4]

[4]Source: https://www.google.com/maps/

Application Fields: Social Media



Current Twitter trending topics in Germany [5]

[5]Source: https://trends24.in/germany/

Solution 1

Prompt Engineering

Prompt Engineering

Definition 2

A prompt is a set of instructions provided to an LLM that programs the LLM by customizing it and/or enhancing or refining its capabilities

Prompt Engineering

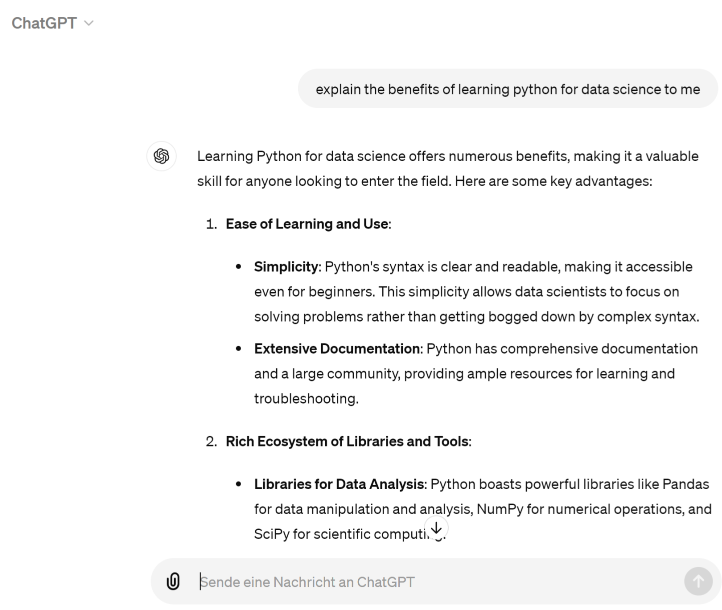
Definition 3

A prompt is a set of instructions provided to an LLM that programs the LLM by customizing it and/or enhancing or refining its capabilities. [13]

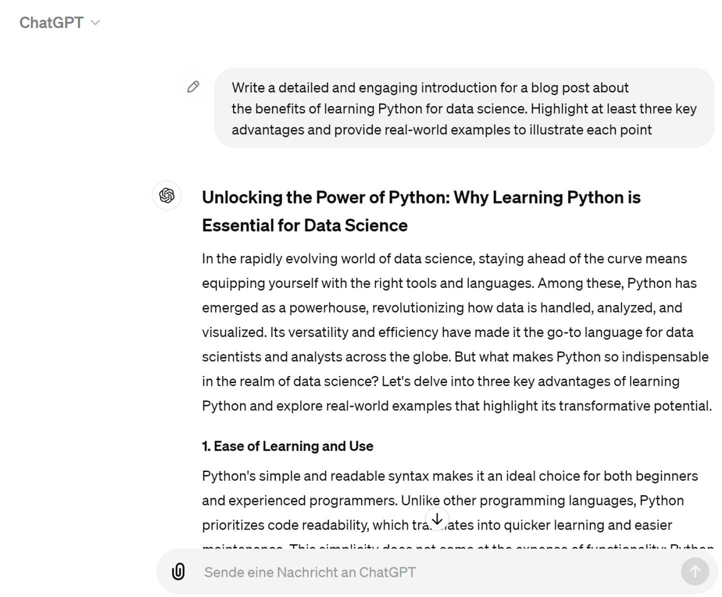
Definition 4

Prompt Engineering is the means by which LLMs are programmed via prompts. [13]

Prompt Engineering



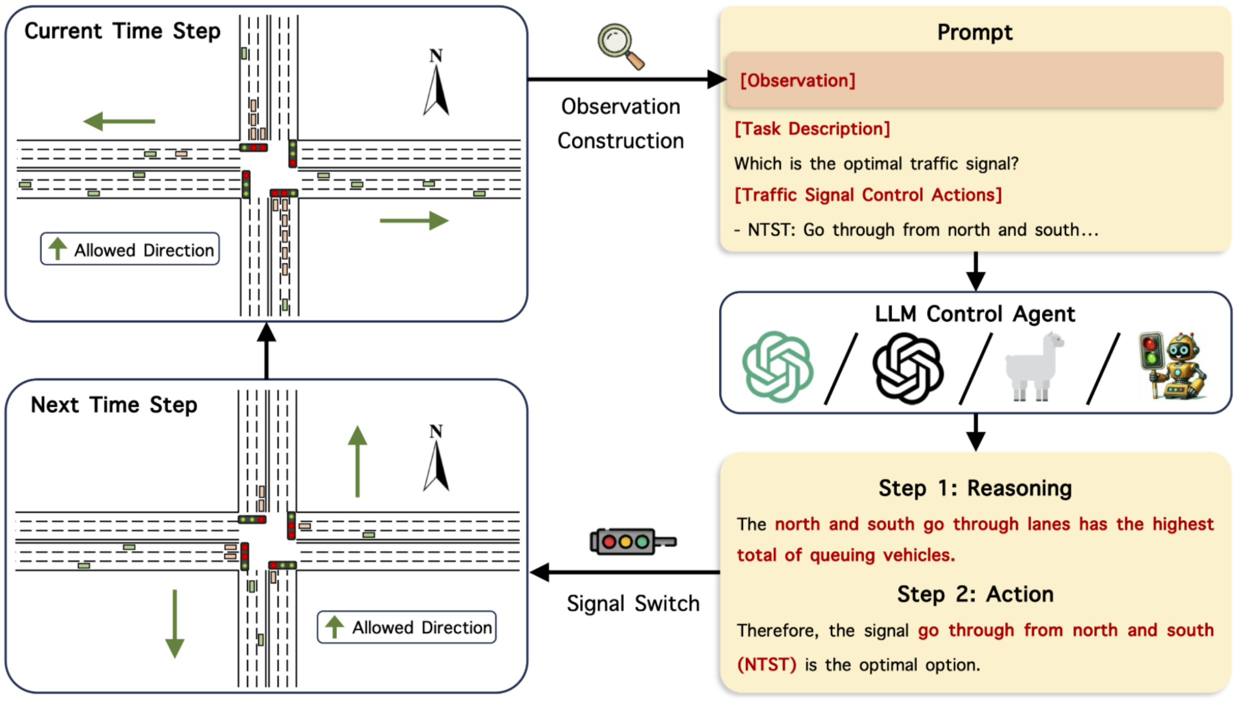
(a) Prompt 1



(b) Prompt 2

2 different prompts yield different output of ChatGPT

Use Case: LLMLight



The workflow of LLMLight [14]

Prompt Engineering

- Convert the real-time data into model-readable prompts and update the model's knowledge via prompt engineering.

Prompt Engineering

- Convert the real-time data into model-readable prompts and update the model's knowledge via prompt engineering.
- However, this method heavily relies on the quality of the prompt formulated by users.

Prompt Engineering

- Convert the real-time data into model-readable prompts and update the model's knowledge via prompt engineering.
- However, this method heavily relies on the quality of the prompt formulated by users.
- A domain specific LLM is needed.

Solution 2

Continual Learning

Continual Learning

Definition 5

continual learning refers to the process of accumulating knowledge on non-stationary data. In the context of large language models, continual learning is applied to enable LLMs to learn from a continuous data stream over time. [15]

Continual Learning

Definition 6

continual learning refers to the process of accumulating knowledge on non-stationary data. In the context of large language models, continual learning is applied to enable LLMs to learn from a continuous data stream over time. [15]

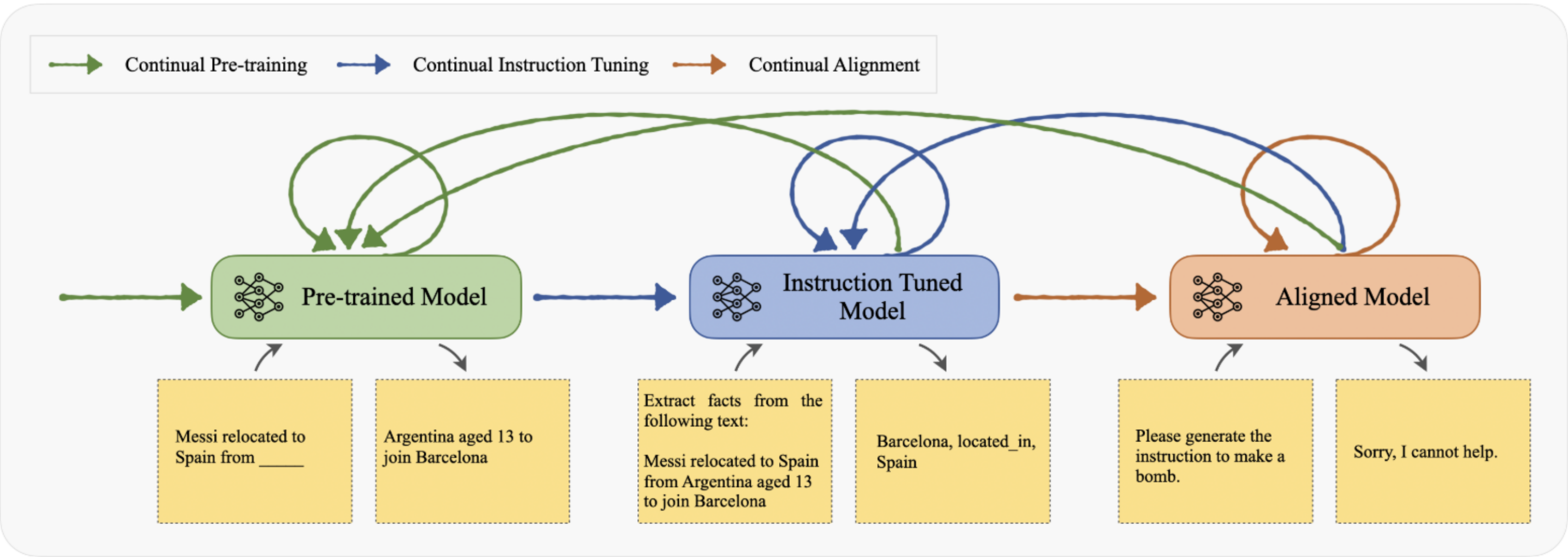
Definition 7

For a data stream of tasks $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$, the goal is to have the model learn sequentially based on the input stream, where it only has access to $\mathcal{T}_i$ at time i. [16]

Stages of Continual Learning[16]

- **Continual Pre-training (CPT)**: It refers to incrementally updating a large language model on input data over time without retraining it from scratch.
- **Continual Instruction Tuning (CIT)**: It refers to continuously refining the model's ability to follow the instruction.
- **Continual Alignment (CA)**: It refers to adjusting the model so it always produces output that satisfies certain standards and guidelines.

Stages of Continual Learning



Stages of continual learning [16]

Continual Learning

However, during continual learning on new input streams, models are likely to forget previously learned knowledge (Catastrophic Forgetting). [2]

Continual Learning

However, during continual learning on new input streams, models are likely to forget previously learned knowledge (Catastrophic Forgetting). [2]

- Replay-based methods
- Regularization-based methods
- Architecture-based Methods

Challenges of Continual Learning [2], [19]

- Catastrophic forgetting
- Lack of real-world assumption
- Multi-modal continual learning
- Concept drift

Conclusion

- Several models such as ChatGPT and GPT4 are not open-sourced, their exact training process and architecture details remain unknown.
- Real-world applications of the two methods are still very limited.
- In reality, data is often irregular and multi-modal, making it costly to transform into a usable format.
- Implement more efficient algorithms to reduce the computational overhead of CA while mitigating catastrophic forgetting
- Develop more effective prompt generation processes.
- Explore more real-world applications.

References

- Liu, Yiheng, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He et al. ”Summary of chatgpt-related research and perspective towards the future of large language models.” Meta-Radiology (2023): 100017.
- Gupta, Kshitij, Benjamin Thérien, Adam Ibrahim, Mats L. Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. ”Continual Pre-Training of Large Language Models: How to (re) warm your model?.” arXiv preprint arXiv:2308.04014 (2023).
- Cavnar, William B., and John M. Trenkle. ”N-gram-based Text Categorization.” In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 161-175. Las Vegas, NV, 1994. Ann Arbor, MI: Environmental Research Institute of Michigan (ERIM).
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. ”Distributed representations of words and phrases and their compositionality.” Advances in neural information processing systems 26 (2013).

References

- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. ”Sequence to sequence learning with neural networks.” Advances in neural information processing systems 27 (2014).
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. ”Attention is all you need.” Advances in neural information processing systems 30 (2017).
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. ”Bert: Pre-training of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 (2018).
- Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun et al. ”Google’s neural machine translation system: Bridging the gap between human and machine translation.” arXiv preprint arXiv:1609.08144 (2016).

References

- Sennrich, Rico, Barry Haddow, and Alexandra Birch. ”Neural machine translation of rare words with subword units.” arXiv preprint arXiv:1508.07909 (2015).
- Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. ”Improving language understanding by generative pre-training.” (2018).
- Geisler, Sandra. ”Data stream management systems.” In Dagstuhl Follow-Ups, vol. 5. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.
- Shi, Haizhou, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. ”Continual Learning of Large Language Models: A Comprehensive Survey.” arXiv preprint arXiv:2404.16789 (2024).
- Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. ”Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.” ACM Computing Surveys 55, no. 9 (2023): 1-35.

References

- Lai, Siqi, Zhao Xu, Weijia Zhang, Hao Liu, and Hui Xiong. ”Large language models as traffic signal control agents: Capacity and opportunity.” arXiv preprint arXiv:2312.16044 (2023).
- Biesialska, Magdalena, Katarzyna Biesialska, and Marta R. Costa-Jussa. ”Continual lifelong learning in natural language processing: A survey.” arXiv preprint arXiv:2012.09823 (2020).
- Wu, Tongtong, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. ”Continual learning for large language models: A survey.” arXiv preprint arXiv:2402.01364 (2024).
- Razdaibiedina, Anastasia, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. ”Progressive prompts: Continual learning for language models.” arXiv preprint arXiv:2301.12314 (2023).

References

- Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. ”Scaling laws for neural language models.” arXiv preprint arXiv:2001.08361 (2020).
- Yang, Yutao, Jie Zhou, Xuanwen Ding, Tianyu Huai, Shunyu Liu, Qin Chen, Liang He, and Yuan Xie. ”Recent Advances of Foundation Language Models-based Continual Learning: A Survey.” arXiv preprint arXiv:2405.18653 (2024).
- Fujii, Kazuki, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. ”Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities.” arXiv preprint arXiv:2404.17790 (2024).