Technical Report: Investigating Independent Data Processing for Custom Data Querying

Ashwin Ramachandran
IIT Bombay
ashwinr@cse.iitb.ac.in

Abstract

Large Language Models (LLMs) have garnered growing significance in the realm of querying personal data, encompassing both structured and unstructured information. Nevertheless, these models confront inherent constraints stemming from their restricted context window size, which impedes the simultaneous inclusion of multiple lengthy documents. In this empirical inquiry, we systematically examine methodologies that enable the independent processing of individual data elements, circumventing the aforementioned constraint. Furthermore, our findings yeild valuable insights into the mechanics by which an LLM handles its input.

1 Introduction

The utilization of Large Language Models (LLMs) has gained prominence in the domain of querying personal data, encompassing queries directed towards private documents or tabular structures within databases (Jiang et al., 2023). Notwithstanding this, a pronounced challenge in this application pertains to the inherent limitation imposed by the finite context window size, constraining the concurrent incorporation of multiple extensive documents (Mialon et al., 2023). In response to this challenge, services such as LlamaIndex (Liu, 2022) have emerged, with the core objective of mitigating this constraint by integrating vector databases with LLMs. This fusion enables the storage of individual data entities in conjunction with their corresponding contextual information within vector repositories, subsequently facilitating retrieval during the query processing phase.

The practice of in-context learning has emerged as a prevalent strategy for adapting pre-trained LLMs to diverse tasks, as discussed in a survey by (Dong et al., 2023). This approach involves furnishing the LLM with multiple instances serving as in-context demonstrations, affording it the capability to discern patterns and respond proficiently to queries. Notably, the presentation of such examples within the context window must occur sequentially to support effective pattern acquisition. However, in the specific context of custom data querying, we find ourselves free from the constraint of sequential ordering. Instead, we possess the flexibility to address each data example in isolation.

This paper embarks on an exploration of techniques geared towards the independent processing of individual data instances, as opposed to their sequential treatment. Our primary objective in this pursuit is to harness greater control over the data under consideration, while concurrently expanding the scale of the data involved. By processing data in an independent fashion, we streamline operations such as data deletion, addition, and modification, without necessitating computational reiterations across all in-context examples.

In the subsequent sections, this paper presents two pivotal contributions:

- A demonstration of the efficacy of various techniques for independently processing data, substantiated by empirical findings.
- Insights drawn from our observations, shedding light on the mechanics governing how an LLM handles sequentially presented data.

It is noteworthy that the techniques expounded upon in this paper demand no structural alterations to the LLMs; they solely pertain to the inference process.

2 Methodology

In our approach, we denote the data as $\{D_i \; \forall \; 0 \leq i < N\}$. $len(D_i)$ represents the number of tokens contained within D_i. Conventional methods typically necessitate that the total token count across all data elements complies with the context window length, denoted as CW, such that $\sum_i^N len(D_i) \leq CW$. However, our objective is to relax this constraint to the condition $len(D_i) \leq CW \ \forall \ 1 \leq i < N$. To achieve this, we adopt a strategy of processing each data element independently, as opposed to a strictly sequential approach.

Our data processing methodology for each D_i is outlined as follows:

- We prepend a limited context, denoted as C to D_i.
- Inference is conducted on each D_i with the context appended, utilizing a transformer model.
- We gather the hidden states corresponding to D_i generated as the output from each layer, resulting in $len(D_i)$ hidden states for each layer.

The "processed" hidden states are subsequently leveraged for inference in response to user queries in the following manner:

- Before initiating the inference process, we modify the key-value cache for each attention head within every layer by undertaking the following steps:
  - The "processed" hidden states are projected to key and value vectors using key and value projection matrices.
  - These computed vectors are incorporated into the corresponding key-value cache.

The subsequent sections of this paper delve into the intricacies of how inference on D_i is executed during the "processing" phase.

2.1 Data Representation D_i

The empirical findings presented in this paper are illustrated through the examination of a specific dataset, which we will denote as "DataSet 1: Person-Action Relationship."

Within this dataset, each D_i corresponds to a concise English sentence encapsulating a person and an associated action, expressed in the format "<Name of Person> is <Name of Action>." Illustrative instances from this dataset include "Leechenbaum is driving" and "Zelensky is hiking."

It is crucial to note that each D_i within this dataset consists of two distinctive entities: a person-entity (referred to as "<Name of Person>") and an action-entity (referred to as "<Name of Action>").

2.2 Naive "Processing" with Empty Context : $C = \emptyset$

In this particular approach, each data element D_i is independently "processed" without the incorporation of any additional contextual information denoted by C. Empirical observations have revealed a significant disparity between the entities present in the data and the manner in which they are interpreted by the Large Language Model (LLM).

For Dataset-1, a notable discrepancy is observed, where the person-entity is frequently associated with an action-entity different from its intended assignment. Table 1 provides a representation of LLM responses to select queries presented to it.

The empirical results demonstrate a substantial misalignment between the "processed" hidden states of the action-entity and their intended encoding of information about the person-entity. Consequently, we deduce that it is imperative to instigate measures to guide the model in encoding this specific information accurately.

Query	**Response**
What is Williamson is doing?	baking
What is Oppenheimer doing?	baking
Who is cycling?	Oppenheimer
Who is baking?	Oppenheimer

Table 1: Data: ["Williamson is baking", "Oppenheimer is cycling"]

2.3 Template Based "Processing": $C = D_0$

In this approach, each data element D_i, where $i > 0$, is augmented with a template resembling the structure of D_i. This augmentation serves to encourage the model to encode pertinent information regarding the identity of the first entity within the hidden states of the second entity.

For Dataset-1, we employ D_0 as the template, consequently modifying the data as follows: "D_0, D_i" for $i > 0$. Tables 2 and 3 showcase the LLM responses to a selection of queries posed to the model.

From table 2, we observe that the responses are as expected. But as we increase the number of data elements to three, the responses begin to be inaccurate as observed in 3.

The empirical results underscore an intriguing observation. The model, under the influence of this template-based "processing" approach, tends to generate mismatches when processing D_i where

$i > 0$. We posit that the model introduces a global positional order to differentiate between distinct data elements. This position is calculated by enumerating the number of separators present in the text being "processed," which, in our specific case, consists of the comma (",") character. Consequently, by manipulating the number of separators in the context, it is feasible to modify the global position assigned to the data.

In Table 3, D_0 is allocated a global position of 0 (in the absence of any separator), while the remaining data elements are assigned a global position of 1 (indicating the presence of exactly one separator). This configuration results in only one of the data elements assigned a global position of 1 being considered for inference.

Query	**Response**
What is Williamson is doing?	baking
What is Oppenheimer doing?	cycling
Who is cycling?	Oppenheimer
Who is baking?	Williamson

Table 2: Data: ["Williamson is baking", "Oppenheimer is cycling"]

Query	**Response**
What is Williamson is doing?	baking
What is Oppenheimer doing?	painting
What is Leechenbaum doing?	cycling
Who is cycling?	Leechenbaum
Who is baking?	Williamson
Who is painting?	Leechenbaum

Table 3: Data: ["Williamson is baking", "Oppenheimer is cycling", "Leechenbaum is painting"]

2.4 Template and Position-Based "processing": $C_i = \{i$ <FLR><SEP> tokens$\} + \{D_0\}$

Each data D_i is prefixed with i <FLR><SEP> tokens along with the template as context. Here, a filler token <FLR> is a group of tokens proportional to the length of the entities present in the data. Empirically, we find that using the first entity present in D_0 gives accurate responses.

For Dataset-1, we use "<FLR>, " $*i$ + "D_0" as the template. Hence, the modified data is now "<FLR>, " $*i$ + "D_0, D_i". In our experiments, we use "Williamson" as the <FLR> token and "," as the <SEP> token. Table 4 and 5 show the LLM responses to some of the queries posed to it.

This "processing" gives us the desired responses for our queries. In the More Results section, we show results for larger number of data elements and larger number of entities per data element.

Query	**Response**
What is Williamson is doing?	baking
What is Oppenheimer doing?	cycling
What is Leechenbaum doing?	painting
Who is cycling?	Oppenheimer
Who is baking?	Williamson
Who is painting?	Leechenbaum

Table 4: Data: ["Williamson is baking", "Oppenheimer is cycling", "Leechenbaum is painting"]

Query	**Response**
What is Williamson is doing?	baking
What is Oppenheimer doing?	cycling
What is Leechenbaum doing?	painting
What is Zelensky doing?	relaxing
Who is cycling?	Oppenheimer
Who is baking?	Williamson
Who is painting?	Leechenbaum
Who is relaxing?	Zelensky

Table 5: Data: ["Williamson is baking", "Oppenheimer is cycling", "Leechenbaum is painting", "Zelensky is relaxing"]

3 Conclusion

From our empirical investigations, we conclude that the LLM, in order to distinguish different pieces of information, uses a global position order, that in our case is determined from the number of seperators present in it's context. If two datum occupy the same global position order, it tends to mismatch the entities present in one with the other.

We also observe that the hidden states of entities do not generally encode information about the previously associated entities. But we can encourage it to do so by supplying a similar datum. This forces the subsequent hidden states to encode more information about entities present in the data since now it has in it's context more than one similar datum.

Using these observations, we demonstrate a technique that can independently process data elements thereby enhancing control over them and eliminating the need for recomputation during user query processing.

4 Future Work

We would like to investigate the possibility of extending the above described technique to different structured and unstructured data such as table schemas and more key-value pairs. Our final desired goal is to generalize our approach to all kinds of data.

Limitations

The empirical investigation has only been conducted for one data set, English sentences describing relationship between multiple entities. More empirical investigations are required to check the generalizability of the approach to other form of data.

The results of the investigations are specific to Llama 2 13b Chat model (Touvron et al., 2023). Further work is required to check whether the same results are observed on other LLMs as well.

References

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew R Gormley. 2023. Unlimiformer: Long-range transformers with unlimited length input. *arXiv preprint arXiv:2305.01625*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation.

Jerry Liu. 2022. LlamaIndex.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Appendix

A.1 Experimental details

We access the Llama-2 13b chat model (Touvron et al., 2023) via huggingface transformers (Wolf et al., 2020). We use the Unlimiformer codebase (Bertsch et al., 2023) for our experiments. We make sure our use of Llama-2 model is within it's licensing agreement.

The system instruction provided to the model: "<s>[INST] «SYS» You are a helpful assistant. Answer with short responses according to the question. «/SYS»".

The queries are asked in the following format: "Based on the above information, can you tell me <INSERT QUERY>?[/INST]"

A.2 More Results

A.2.1 Importance of Template D_0 in the Context C

Results in Table 6 and Table 7 demonstrate the importance of template supplied in the context.

With a smaller template such as "Williamson is baking", we get inaccurate results. But by using D_0 which is "Williamson is baking in America", we get accurate results.

A.2.2 Importance of <FLR> token in the Context C

Results in Table 7 and Table 8 demonstrate the importance of token supplied in the context.

Query	**Response**
What is Williamson is doing?	baking in America
What is Oppenheimer doing?	cycling in Lithuania
What is Leechenbaum doing?	painting in Siberia
What is Zelensky doing?	relaxing in India
What is Murugan doing?	eating in Siberia

Table 6: Data is ["Williamson is baking in America", "Oppenheimer is cycling in Lithuania", "Leechenbaum is painting in Siberia", "Zelensky is relaxing in India", "Murugan is eating in Mexico"]; context supplied is ($C_i = i*$"Williamson"+"Williamson is baking")

Query	**Response**
What is Williamson is doing?	baking in America
What is Oppenheimer doing?	cycling in Lithuania
What is Leechenbaum doing?	painting in Siberia
What is Zelensky doing?	relaxing in India
What is Murugan doing?	eating in Mexico

Table 7: Data :["Williamson is baking in America", "Oppenheimer is cycling in Lithuania", "Leechenbaum is painting in Siberia", "Zelensky is relaxing in India", "Murugan is eating in Mexico"]; context supplied is ($C_i = i*$"Williamson"+"Williamson is baking in America")

Using a smaller <FLR> such as "Lory" gives poor results. Using a entity-size proportional <FLR> such as "Williamson" gives accurate results

Query	**Response**
What is Williamson is doing?	baking in America
What is Oppenheimer doing?	cycling in Lithuania
What is Leechenbaum doing?	cycling in Siberia
What is Zelensky doing?	eating in India
What is Murugan doing?	eating in Siberia

Table 8: Data is ["Williamson is baking in America", "Oppenheimer is cycling in Lithuania", "Leechenbaum is painting in Siberia", "Zelensky is relaxing in India", "Murugan is eating in Mexico"]; context supplied is ($C_i = i*$"Lory"+"Williamson is baking in America")