Attention Overflow: Language Model Input Blur during Long-Context Missing Items Identification

Anonymous ACL submission

Abstract

Large language models (LLMs) can suggest missing elements from items listed in a prompt, which can be used for list completion or similar item recommendation. However, their performance degrades when they are exposed to too many items, as they start to suggest items already included in the input list. This occurs at around 100 items for mid-2024 flagship LLMs. We evaluate this phenomenon on both synthetic problems (e.g., finding missing numbers in a given range of shuffled integers) and realistic movie recommendation scenarios. We refer to this issue as *attention overflow*, as avoiding repetition requires attending to all items simultaneously. Although iterative loops can mitigate this problem, their costs increase with the repetition rate, affecting the language models' ability to derive novelty from lengthy inputs.

1 Introduction

Large language models (LLMs) boast ever-growing context windows, enabling new potential applications. However, the theoretical context length is not a sufficient indication of a model's real performance with a given input size (Liu et al., 2024). Multiple benchmarks have been proposed to stress-test the actual ability of language models to reason over long contexts. These tasks either involve pure retrieval or a form of reasoning requiring the identification of a few relevant portions from a large context.

We question the effective context window of language models from an opposite angle: asking them to provide the only relevant elements that are *not* in a large input. We formulate this as a missing item prediction task. Missing item prediction has multiple applications, notably in conversational recommendation, where users can provide a list of items (e.g. movies) they liked and ask for new suggestions. This task involves a form of inductive reasoning, in contrast to the deductive reasoning typically explored in long context stress tests. More importantly, it requires comparing a representation to the whole input, and we notice that this is difficult for current LLMs, which leads to the prediction of items already in the input (repetition).

Missing item prediction is also relevant when models are asked to generate long lists. We observed repetitions in this scenario[1], but we focus on the movie recommendation use case, where users provide the movies they have watched, and we also create synthetic examples, notably number ranges with a missing element. We quantify the repetition phenomenon with existing off-the-shelf language models and investigate whether fine-tuning can easily address this problem. The created datasets are publicly available[2].

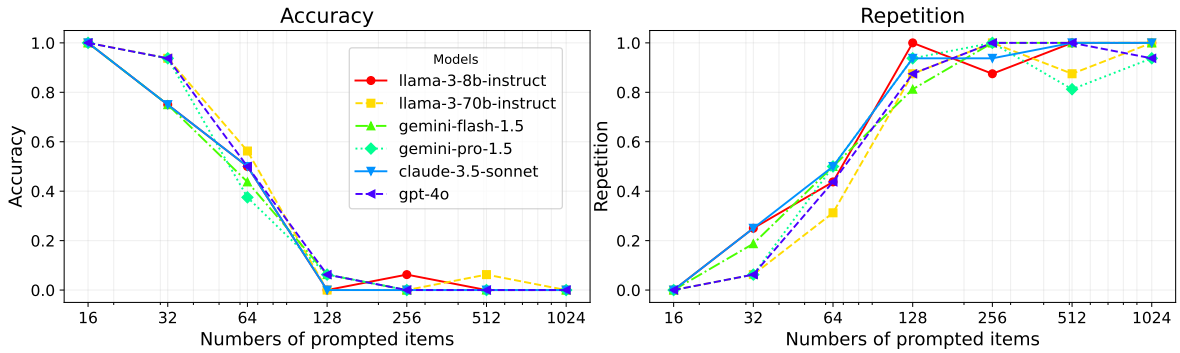
2 Related work

Repetitions in language modeling We study a form of repetitions, a well-identified problem in language models (Keskar et al., 2019), which can sometimes lead to text degeneration, where models repeat the same token indefinitely (Fu et al., 2021). Repetition penalties were proposed to alleviate this issue (Keskar et al., 2019), but they operate at the token level and cannot scale to large contexts where all tokens are already represented. Repetitions also exist in more subtle ways, as Chiang and Lee (2024) showed that chain-of-thought reasoning contains redundant content.

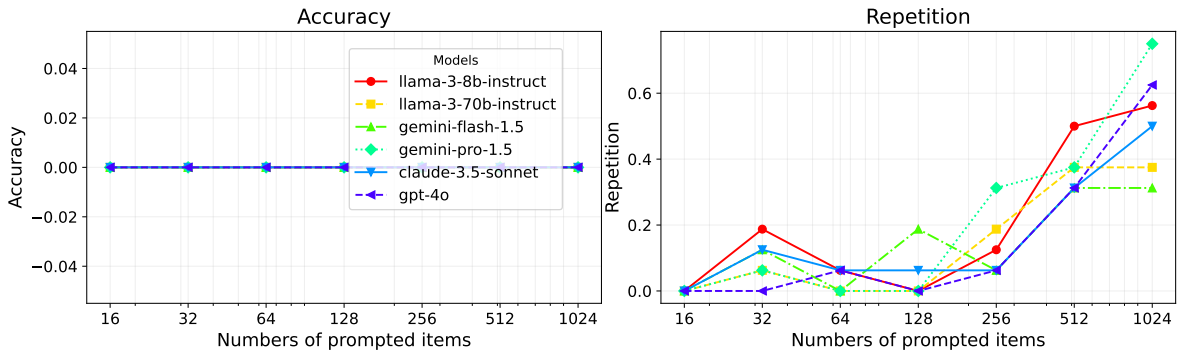
LLM context length stress tests Our work is also related to context window stress testing and language modeling-based recommendation. Previous work has studied the ability of attention mechanisms to identify what is present in long contexts, but not what is missing. The Long-Range Arena (Tay et al., 2021) provides the first systematic analy-

[1] For example, asking Claude Sonnet 3.5 200 movies released in 2022 leads to numerous repetitions: [artifact]

[2] [data:HF-datasets 🤗]



(a) Zero-shot missing number prediction



(b) Zero-shot missing movie prediction

Figure 1: Zero-shot test accuracy and repetition rate with increasing itemset sizes.

sis of the long-range processing capabilities of text encoders, focusing mainly on algorithmic reasoning and retrieval tasks. BABILong (Kuratov et al., 2024) uses bAbi reasoning tasks (Weston et al., 2016) and interleaves relevant text with irrelevant input. FlenQA (Levy et al., 2024) applies a similar process to the RuleTaker (Clark et al., 2020) deductive logical reasoning task. Ruler (Hsieh et al., 2024) uses simple algorithmic/retrieval tasks.

Recommendation with LLMs Our study is also related to LLM usage for collaborative filtering (Sileo et al., 2022), where users enumerate a list of items to communicate their tastes. LLMs can also be used in content-based recommendations, where users explicitly mention what they are looking for (Wu et al., 2023). Here, we do not address the fine-grained relevance of the recommendations (providing an item that users do not already know). Repetition is also related to the novelty metric in recommender systems evaluation (Vargas and Castells, 2011).

3 Missing item prediction

We formalize the task of missing item prediction as follows: Given a set X (randomly shuffled) of N elements, guess the element y that is missing in X. This is technically an induction task that can be under-determined but we can construct relatively easy X, y pairs with easily identifiable itemsets $\mathcal{S}$ (numbers from 0 to 1024, letters, chemical elements...) and randomly removing one element y from $\mathcal{S}$ to get X. We can use two evaluation metrics:

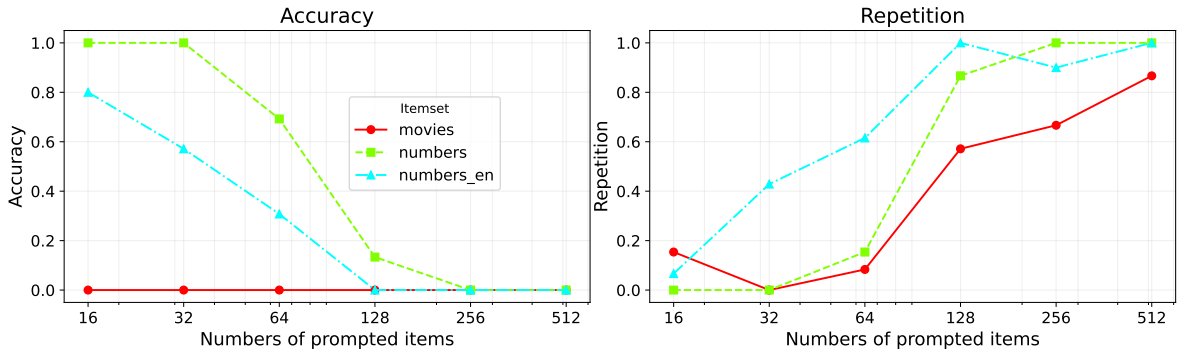
Accuracy the rate at which a language model returns the expected missing element.

Repetition rate the rate at which a language model returns an element that is already in X.

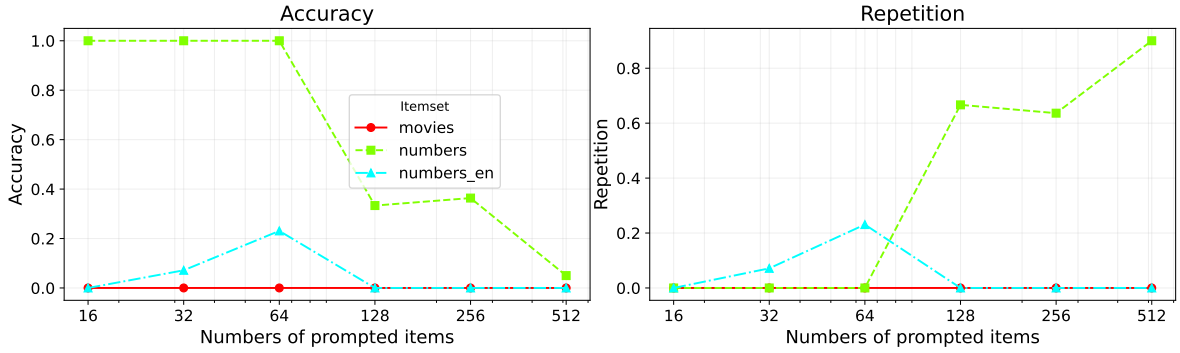
Repetitions are always mistakes. For easily identifiable sets, ideal behavior is perfect accuracy and no repetition. But even in cases where the structure of $\mathcal{S}$ is under-determined, language models performing missing item prediction should not repeat elements from X.

To construct an example of the missing item prediction task, we select an itemset $\mathcal{S}$, select a random element y, and present a scrambled version of $X = \mathcal{S} \setminus \{y\}$ in a prompt explicitly asking the model to guess a missing element. We provide the following itemsets:

Movies We select a user from the MovieLens 1M dataset who watched more than 2048 movies.



(a) Llama-3 zero-shot missing number prediction on multiple domains



(b) Llama-3 fine-tuned on missing number prediction

Figure 2: Llama-3-8B-Instruct Accuracy on various itemsets with increasing itemset sizes, without any fine-tuning (a) and after fine-tuning on the numbers itemset.

Numbers Numbers in numerical form (1...1024). We exclude set extrema from the choice of y for numerical itemsets.

Numbers-english We use the same numbers but converted in English using the num2word library[3].

An example with the Numbers itemset of size 8 is QUESTION: *Find the missing element in 5, 7, 1, 3, 6, 8, 4.* ANSWER: 2.

4 Experiments

We use the same prompt template for all models:

> Guess the missing item from this list: {X}. Directly answer with only one item. Item format should match the list format. Provide no explanation. Answer format: "{item}."

To construct this prompt template, we iterated on Llama-3-8B-Instruct with the numbers itemset validation data until we obtained a satisfactory output format. We normalize the outputs with punctuation removal and lowercasing to compute repetition rate and accuracy, and perform exact matches to compute accuracy and repetition rate.

[3] https://github.com/savoirfairelinux/num2words

We use powers of 2 starting from 16 as itemset sizes. This ensures that there are enough items to guess the itemset structure. We generate 200 train examples and 100/100 validation test examples per itemset size and itemset type.

4.1 Zero-shot evaluation

We evaluate off-the-shelf instruction-tuned language models API through OpenRouter. We evaluate Llama3-Instruct 8B and 70B, Gemini 1.5 Flash and Pro, GPT-4o, and Claude 3.5 Sonnet with the default hyperparameters.

Figure 1 shows the evolution of Accuracy and Repetition metrics with different itemsets sizes for numeric numbers and movies missing item prediction tasks. Most language models solve the missing number prediction task with relatively high accuracy with less than 128 items. Increasing model size improves accuracy, as Gemini Pro and Llama-3-70B outperform their smaller counterparts. However, the repetition rates shoot up and the accuracy decreases in all models after 256 items.

We cannot interpret the low accuracy of the movie item prediction tasks as a failure because the models can predict relevant movies that are not

y. However, we can interpret the growing repetition rate as a failure, which can frustrate users who could expect better recommendations as they provide more examples, which limits the accuracy of conversational recommender systems that do not filter their output to prevent repetitions.

4.2 Fine-tuning

We now investigate whether fine-tuning can easily address this issue. We fine-tune Llama-3 Instruct 8B using Unsloth default configuration [4] (4bit quantization, LoRA (Dettmers et al., 2024) with dimension 16, 1 epoch with a learning rate of 2e-4). We fine-tune on 500 numeric items of a size below 256 and evaluate on the test set in-domain and out-domain.

Figure 2 shows that fine-tuning improves missing item prediction on in-domain data, but do not generalize to larger itemsets nor to different domains, which might indicate a fundamental limit of current attention architectures that may not be solved with data only.

4.3 Contrastive evaluation

We also evaluated the ability of LLama-3-8B-Instruct to tell whether an element is present or not in the list by randomly sampling either the missing element or a random element from a prompt.

> {X}. Is "{item}" in the previous list? Provide no explanation, directly answer with only "Yes." or "No."

Figure 3 shows the evolution of accuracy with growing itemset sizes. Llama-3-8B-Instruct maintains 75% accuracy below 1024 items[5]. This shows that once the item is explicitly present in the query, the model is much better at identifying it. These results are lower than the Needle in a Haystack evaluation scores of Llama-3 (Zhang et al., 2024), which is due to the high similarity between items. This suggests that context-length stress testing is harder when many prompt elements are similar to each other, which makes existing (Kuratov et al., 2024) problem lengthening strategies too easy to get around.

[4] https://colab.research.google.com/drive/135ced7oHytdxu3N2DNe1Z0kqjyYIkDXp?usp=sharing

[5] All examples fit in the 8K context window of Llama 3.

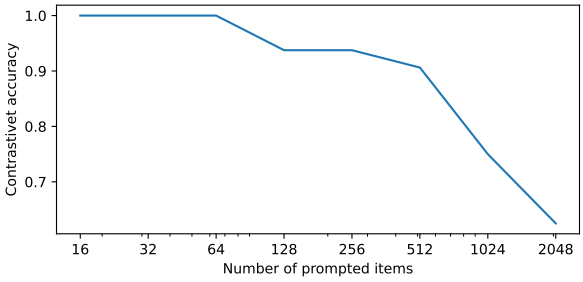


Figure 3: Zero-shot contrastive accuracy with Lllama-3-8B-Instruct on the Numbers itemset.

5 Analysis

To solve missing item prediction, a transformer language model needs to construct a latent representation of the missing item when predicting the next token. Finding a close representation is relatively simple in the tasks we propose, as language models consistently output items that belong to the item set. However, they also need to move away the latent representation from the representation of each prompted item. At each layer, the transformer layer can refine the representation to shift it away from prompted items. Our contrastive experiments show that it is possible with relatively high accuracy when evaluating one item at a time, but the models lack the depth to do it for many items.

6 Conclusion

We introduce a new missing item prediction dataset and we show that repetitions occur during plausible movie recommendation tasks, alongside synthetic list completion. Our findings have implications on the current language models' ability to check text exhaustivity. Our simple examples show that we must be careful when asking language models to produce new content from contextual information, as language models can repeat context elements without noticing it. Our research demonstrates that issues arise with sequences of just hundreds of tokens, in contrast to the RULER (Hsieh et al., 2024) study, which identified problems only at lengths of thousands of tokens. We attribute this phenomenon to an overflow of attention, speculating that the model needs to evaluate candidates and compare them to all input items at once. It would be worthwhile to actually analyze the attention heads during this task, even though multi-head attention is hard to interpret (Bibal et al., 2022). Our dataset is publicly available with itemset sizes up to 8192 for future work.

Limitations

Our study has several limitations that should be considered. The range of itemsets and models tested could be expanded to ensure broader generalizability. Our fine-tuning experiments were limited in scope, and a more systematic approach to prompt engineering might yield different results. We also lack a human baseline for comparison. The movie recommendation task is a simplified version of real-world scenarios, which often involve more complex user preferences and item attributes. While we speculate about the role of attention in the observed phenomenon, we did not conduct an in-depth analysis of attention patterns or explore how efficiently the models utilize their full context window. Addressing these limitations in future work would provide a more comprehensive understanding of the attention overflow phenomenon and its implications for long-context processing in language models.

References

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. 2022. Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900.

Cheng-Han Chiang and Hung-yi Lee. 2024. Over-reasoning and redundant calculation of large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–169, St. Julian's, Malta. Association for Computational Linguistics.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization. Main track.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2021. A theoretical analysis of the repetition problem in text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 14, pages 12848–12856.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *arXiv preprint arXiv:2406.10149*.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *arXiv preprint arXiv:2402.14848*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Damien Sileo, Wout Vossen, and Robbe Raymaekers. 2022. Zero-shot recommendation as language modeling. In *Advances in Information Retrieval*, pages 223–230, Cham. Springer International Publishing.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 109–116.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomás Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*.

Peitian Zhang, Ninglu Shao, Zheng Liu, Shitao Xiao, Hongjin Qian, Qiwei Ye, and Zhicheng Dou. 2024. Extending llama-3's context ten-fold overnight. *arXiv preprint arXiv:2404.19553*.