What Kinds of Tokens Benefit from Distant Text? An Analysis on Long Context Language Modeling

Anonymous ACL submission

Abstract

As the context length that large language models can handle continues to increase, these models demonstrate an enhanced ability to utilize distant information for tasks such as language modeling. This capability contrasts with human reading and writing habits, where it is uncommon to remember and use particularly distant information, except in cases of foreshadowing. In this paper, we aim to explore which kinds of words benefit more from long contexts in language models. By analyzing the changes in token probabilities with increasing context length, we find that content words (e.g., nouns, adjectives) and the initial tokens of words benefit the most. Frequent patterns in the context (N-grams) also significantly impact predictions. Additionally, the model's prior knowledge plays a crucial role in influencing predictions, especially for rare tokens. We also observe that language models become more confident with longer contexts, resulting in sharper probability distributions. This overconfidence may contribute to the increasing probabilities of tokens with distant contextual information. We hope that our analysis will help the community better understand long-text language modeling and contribute to the design of more reliable long-context models.

1 Introduction

Many studies have expanded the context window of Large Language Models (LLMs) to process longer inputs, leading to the development of long-context LLMs (Zhu et al., 2023; Chen et al., 2023c; Ding et al., 2023a; Peng et al., 2023a). To explore whether a model can effectively process extremely long texts, the language modeling task is frequently evaluated. This task is straightforward to calculate and can be extended to inputs of any length. Language modeling tests the accuracy of predicting the next word based on the previously input text. Previous research has found that as the context length increases, the model's performance in language modeling improves, indicating that the model can utilize more distant information.

When examining models' performance on long-context language modeling tasks, we discover an interesting phenomenon: as the context length increases, the model's language modeling ability continues to improve, even when the input text is very long, such as 32k tokens. This phenomenon does not align with human writing and reading habits. Except for cases involving foreshadowing, it is rare for people to recall particularly distant information in everyday writing or reading, let alone use such distant information to assist in their subsequent writing. This may be because humans, when solving problems, are limited by their working memory and find it difficult to remember too much information simultaneously (Cowan, 2010).

In long-context language modeling tasks, we are curious (1)*Does only a small number of words benefit from ultra-long contexts, just like humans perform foreshadowing?* If not, (2)*what kinds of tokens benefit from the additional distant text in long-context LLMs' language modeling?*

To answer this question, we compare the tokens' probabilities among different context lengths. We find that, when context length increases, there are more tokens whose probability scores increase while fewer whose scores decrease. This indicates a large part of tokens can benefit from the additional distant text, different from human habits that only a few tokens are related to distant foreshadowings. So we wonder, what kinds of tokens benefit from the additional distant text in long-context LLMs' language modeling? To answer this question, we plan to analyze it from three aspects: the characteristics of the **words** themselves, the **context** in which the tokens appear, and the **priors** inherent in large language models.

We analyzed the changes in the predicted probabilities of different tokens as context length increases, using Longlora, Yarn, and Yi as representative long contexts. The greater the increase in a token's predicted probability with increasing context length, the more we consider that token's prediction to be influenced by the long context. By comparing the changes in predicted probabilities for different types of tokens, we derived the following conclusions:

(1). From the perspective of the words themselves, we found that **content words such as nouns and adjectives benefit more from longer contexts.** On the other hand, in current large models, words are often split into several tokens. We discovered that the first token of a word is more influenced by the length of the context, while the predicted probabilities of subsequent subwords remain relatively stable. (2). From the perspective of the context, **frequent patterns in long texts, i.e., N-grams, have a significant impact on the prediction of the next word.** We observed that if the additional context frequently provides an N-gram containing the token to be predicted, the model can predict this token with a higher probability. (3). We also found that **the model's prior has an influence, though possibly not as significant as the impact of the context.** We observed that if a token appears very frequently in the pre-training corpus, meaning the model has a strong grasp of this token, its predicted probability is less likely to be affected by the length of the context. However, for tokens that are rare in the training corpus, their predicted probabilities are more likely to change with the context length. The above conclusions hold only when the impact from the frequent patterns does not change as the context length increases. Otherwise, the probability changes are mainly influenced by the context and have no significant correlation with the model's prior.

We also find that, apart from the tokens aforementioned benefit from the distant text, tokens that are incorrectly predicted by LLMs also show a higher average probability. We speculate that long-context LLMs are more confident as the context length increases. Specifically, with longer input, the probability distribution predicted by LLMs becomes sharper, i.e., the max probability becomes larger, regardless of whether the model correctly predicts the token. Therefore, overconfidence may be one of the reasons that perplexity continuously decreases as the context length increases.

To summarize, the contributions are as follows:

1) We find the inconsistency between humans' behavior and LLMs' performance on long-context language modeling. Humans are unlikely to use distant information for their subsequent writing, while for long-context LLMs, a large number of tokens can benefit from distant text to have a higher probability.

2) We analyze what kinds of tokens are more likely to benefit from distant context from aspects of the characteristics of the **words** themselves, the **context** where the tokens appear, and the **priors** inherent in LLMs, providing insights for a better understanding of long-context LLMs.

3) We find the growing confidence of LLMs as the context length increases, which may be one explanation for the continuously decreasing perplexity. Thus, researchers should carefully consider the impact of overconfidence to design more reliable long-context LLMs.

2 Preliminary

2.1 Perplexity

Perplexity is widely used to evaluate language models. As it can be calculated on any text without length limitation, perplexity is widely used for LLMs' long context processing ability evaluation (Press et al., 2021; Ding et al., 2023b; Zhu et al., 2023). Following Press et al. (2021), we use the sliding window evaluation of perplexity in evaluating long-context LLMs. Specifically, given M documents $\{D_1, ..., D_M\}$ and context length K, long-context LLMs predict each token x_i^t in each document D_t based on previous K tokens when $i \geq K$, otherwise, based on all its previous tokens:

$$p_K(x_i^t) = P(x_i^t|x_0^t, ..., x_{i-1}^t), i \in [0, K) \quad (1)$$

$$p_K(x_i^t) = P(x_i^t|x_{i-K}^t, ..., x_{i-1}^t), i \in [K, |D_t|) \quad (2)$$

We illustrate the sliding window evaluation on the left of Figure 1.After obtaining all tokens' probabilities in D_t, perplexity (PPL) is calculated in the same way as Chen et al. (2023c):

$$\bar{p}_K(x_i^t) = -log(p_K(x_i^t)), i \in [0, |D_t|) \quad (3)$$

$$PPL_K = \frac{1}{M}\sum_{t=1}^{M}(\frac{1}{|D_t|}\sum_{i=0}^{|D_t|-1}\bar{p}_K(x_i^t)) \quad (4)$$

where we denote $\bar{p}_K(x_i^t)$ as **token-perplexity** in our paper. According to the definition of perplexity, the lower perplexity indicates a better language modeling ability[1].

2.2 Perplexity Decreases as Context Length increases

Recently, many LLMs have been proposed with the ability to handle extremely long context. For example, GPT-4 (OpenAI, 2023) has a context window of 128k, while Yi (AI et al., 2024) can even process 200k tokens.

When using perplexity to evaluate the language modeling ability of long-context LLMs, a vast majority of previous works (Press et al., 2021; Zhu et al., 2023; Chen et al., 2023c; Peng et al., 2023b) unanimously report that perplexity decreases as the context length K increases. This indicates the model can understand the entire document better (Zhang et al., 2023). Such a phenomenon still exists when K scales up from 32k to 64k (Chen et al., 2023c; Peng et al., 2023b). However, this phenomenon does not align with human writing habits. With limited working memory capabilities, it is difficult for humans to remember all information that is too far away (Cowan, 2010). Therefore, except for foreshadowings, humans may seldom refer to distant information when writing subsequent words. Words in the document are more likely to be related to local text. So we wonder if the models perform like humans so that the decrease in perplexity comes from a few tokens, such as tokens related to foreshadowing. If not, this indicates that not only tokens related to foreshadowings, but also some other tokens written based on local information by humans can benefit from the distant text by long-context LLMs, which provokes us to wonder, what kinds of these tokens are?

3 Experimental Setup

To answer the aforementioned questions, we need to have a detailed analysis regarding the token-perplexity of each token in documents predicted by long-context LLMs.

Models We select three representative long-context LLMs for experiments, which use different methods to scale up the context window to more than 100k. 1) Yi-6B-200K (AI et al., 2024), which adjusts the base frequency of position embedding Rotary Position Embeddings (RoPE) (Su et al., 2023a) for context window extension. 2) YaRN-7B-128K (Peng et al., 2023a), which extends RoPE by interpolating frequencies unevenly and keeping high frequencies intact. 3) LongLoRA-7B-100K (Chen et al., 2023c), which proposes shifted sparse attention (S2-Attn) to approximate long context learning while retaining the original attention architecture during inference.

[1]Please refer to Press et al. (2021) for more details about sliding window perplexity evaluation.

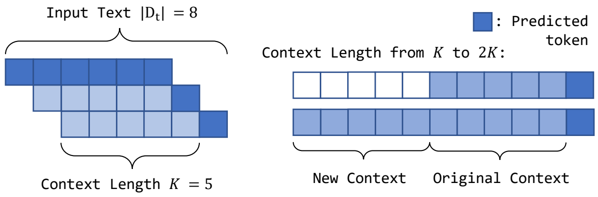

Figure 1: Left part: an illustration for sliding window method of perplexity calculation. Right part: an illustration of *original context* and *new context*.

Dataset Following (Chen et al., 2023c), we calculate perplexity on PG-19 (Rae et al., 2019), a language modeling benchmark, including a set of books derived from the Project Gutenberg books project[2]. Due to the computation resources limitation, we randomly sample 6 books from the test set of PG-19 as our test corpus for experiments. The lengths of selected books are all larger than 64k after tokenization.

Setup First of all, the sliding window evaluation method mentioned in 2 is time-consuming because every time LLMs predict a token, the input text needs to be re-encoded. Following Press et al. (2022), we make predictions on S tokens instead of one every inference pass to improve the computational efficiency. Please refer to Appendix A for more details about *stride* S.

Secondly, we use the change of token-perplexity when context length increases to determine whether the token benefits from the additional text. Specifically, we compare the token-perplexity between the context length of K and $2K$ in our experiment. The illustration is shown in the right part of Figure 1. For simplicity, we name the input context when context length is K as "*original context*" $[x^t_{i-K+1}, ..., x^t_{i-1}]$, while naming the newly added context when context length is extended to $2K$ as "*new context*" $[x^t_{i-2K+1}, ..., x^t_{i-K}]$ in our paper. When the token-perplexity decreases, we say that the token can benefit from the *new context*.

[2]https://www.gutenberg.org

K		2k	4k	8k	16k	32k
Decrease Ratio	Yi	52.6%	52.0%	49.2%	45.8%	39.8%
	YaRN	55.0%	53.0%	51.4%	49.0%	44.5%
	LongLoRA	54.6%	54.1%	54.7%	50.1%	45.3%
Increase Ratio	Yi	43.7%	43.8%	41.8%	39.8%	35.4%
	YaRN	45.0%	45.3%	45.2%	44.2%	41.8%
	LongLoRA	45.3%	45.8%	45.1%	43.1%	41.0%

Table 1: Ratio of tokens with token-perplexity decrease and increase.

K	2k	4k	8k	16k	32k	64k
Yi	2.105	2.078	2.055	2.037	2.024	2.015
YaRN	1.944	1.918	1.896	1.880	1.867	1.859
LongLoRA	2.084	2.051	2.022	2.001	1.985	1.976

Table 2: Perplexity on test corpus.

4 Most Tokens' Token-perplexity Decrease

We first verify that the phenomenon that perplexity decreases as K increases still exists in our experimental setting. In Table 2, we report the perplexity of three long-context LLMs with K ranging from 2k to 64k. The results show that perplexity consistently decreases as K increases.

Therefore, we want to figure out whether the decrease in perplexity comes from a uniform decrease in most token-perplexity or from a drastic decrease in some token-perplexity. We count the proportion of tokens with a decreased token-perplexity to the total number of tokens when the context length is extended from K to $2K$.

The experimental results are shown in Table 1. The sum of each decrease ratio and its counterpart is less than 100% since some token-perplexity remains unchanged. In almost all settings, the token-perplexity decrease ratios are larger than 40%, and consistently surpass the increase ratio as well. Therefore, we can conclude that the decrease in perplexity comes from a decrease in the token-perplexity of a large number of tokens, which contradicts human habits that most tokens are only written with local information.

5 What Tokens Benefit from Distant Text

To investigate this issue, we conduct the analysis from three perspectives: characteristics of the **words** themselves, the **context** where the tokens appear, and **priors** inherent in long-context LLMs.

5.1 Properties of Words

Lexical Property. Function words, such as adpositions that play a grammatical role, and content words (semantically richer words), such as proper nouns that contain more information, are two major groups of parts of speech (POS). Bell et al. (2009) mentions that function words are much more frequent and predictable than content words. As content words contain more information while function words rarely change form or meaning in different contexts, we wonder, provided with additional text, whether content words can benefit more than function words as content words may be more related to global information. Specifically, we want to investigate the correlation between the types of POS and the decreased value of token-perplexity in longer context length.

To determine the POS of each token x_i, we first find its corresponding word w_i in the document before tokenization. Then we use Stanford CoreNLP (Manning et al., 2014) to obtain the POS of w_i, and we treat it as the POS of token x_i as well. According to the definition of the POS tag set [3,4], we divide them into six main classes: *noun*, *verb*, *adj*, *adv*, *closed* (such as adposition, particle, pronoun, etc.) and *other* (not a word, such as punctuation, symbols, etc.). Among the six classes, *noun*, *verb*, *adj*, and *adv* are called "open words" (content words), which usually contain important information, while *closed* means "closed words" (function words), which are used to stitch "open words" together.

To compare the token-perplexity between the context length of K and $2K$, we calculate the **token-perplexity decrement** of each token x_i as:

$$\Delta\bar{p}_K^{2K}(x_i) = -(\bar{p}_{2K}(x_i) - \bar{p}_K(x_i)) \qquad (5)$$

The larger $\Delta\bar{p}_K^{2K}(x_i)$, the larger decrement of token x_i's token-perplexity. For brevity, we abbreviate it as $\Delta\bar{p}(x_i)$

We further calculate the average token-perplexity decrement in each class of POS tags on the test corpus, with *original context* length K ranging from 2k to 32k. The results are shown in Figure 2. We can observe that "*noun*" and "*adj*" deliver the largest token-perplexity decrement in all context length K while tokens in "*closed*" and

[3] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[4] https://web.stanford.edu/~jurafsky/slp3/slides/8_POSNER_intro_May_6_2021.pdf

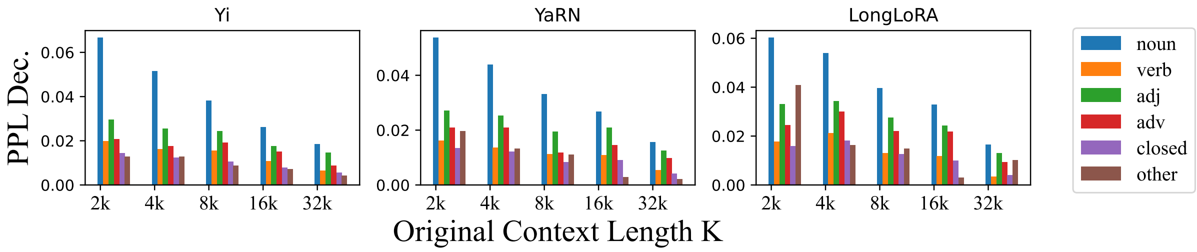

Figure 2: The average token-perplexity decrement in each class of POS tags.

"*other*" show the minimal decrease in most cases. These results indicate the additional information in *new context* benefits "*noun*" and "*adj*" most.

5.1.1 Structures inside Words

Apart from examining tokens that are easily influenced by *new context* from a lexical perspective, we can also consider the structure inside the word itself. Specifically, after tokenization, a word w_j may be split into multiple tokens $[x_{w_j,0}, ..., x_{w_j,n}]$. The first token $x_{w_j,0}$ can only be predicted based on previous words while subsequent tokens can be predicted by leveraging previous tokens in the same word. Therefore, we wonder whether LLMs need more information to infer the starting token $x_{w_j,0}$, that is, whether the first token of a word will benefit more from the *new context*, resulting in a larger decrement in token-perplexity.

To evaluate the relationship between the tokens' position in words, we compare the token-perplexity decrement $\Delta\bar{p}(x_i)$ of the first tokens $x_{w_j,0}$ and the later tokens $x_{w_j,i(i\neq 0)}$ in words. Specifically, we separate all tokens in the test corpus into two sets: **Fir**(st) and **Lat**(ter) according to whether the token is the first token of its original word. Then we calculate the average token-perplexity decrement of each set and compare the decrement value between two sets. The difference between the token-perplexity decrement of the two sets is computed as:

$$\Delta D = \frac{1}{|Fir|}\sum_{i\in Fir}\Delta\bar{p}(x_i) - \frac{1}{|Lat|}\sum_{i\in Lat}\Delta\bar{p}(x_i) \tag{6}$$

Considering POS classes are an important factor of token-perplexity decrement, we calculate ΔD for each class of POS. The results are shown in Figure 3. The results of class "*other*" are not reported because all words in "*other*" can not be tokenized into multiple tokens. Except for "closed", ΔD of all classes of POS is larger than 0, which demonstrates that long-context LLMs deliver a larger token-perplexity decrement in the first tokens with additional text. We can also observe that, the first tokens of POS classes "*noun*" and "*adj*" decrease more than other POS classes in most cases. Therefore, the first token of open words, especially for POS classes "*noun*" and "*adj*", can benefit most from the *new context*.

K	**2k**	**4k**	**8k**	**16k**	**32k**
Yi	0.530	0.347	0.417	0.410	0.356
YaRN	0.342	0.455	0.461	0.349	0.345
LongLoRA	0.380	0.408	0.344	0.340	0.307

Table 3: Correlation coefficients between the token-perplexity decrement $\Delta\bar{p}$ and the N-gram's new occurrence ratio $\Delta\mathcal{N}$. All reported correlation coefficients have p-values < 0.005.

5.2 Influence of Context

Effect of N-gram's Occurrence. If a phrase frequently appears in the context, LLMs are more likely to pay attention to this phrase. Therefore, when given the first few words of the phrase, LLMs may attend to the phrase which appears multiple times in the previous context, and predict the latter words accurately. From this perspective, we want to figure out whether the more an N-gram appears in the input text, the more possible the long-context LLMs refer to the N-gram for prediction, i.e., whether there is a correlation between the token x_i's token-perplexity and the number of the token's N-gram $g_i = [x_{i-N+1}, ..., x_i]$ occurrences.

We first count the number of N-gram g_i^t's occurrences in the *original context* $[x_{i-K+1}^t, ..., x_{i-1}^t]$ and the *new context* $[x_{i-2K+1}^t, ..., x_{i-K}^t]$ of document D_t, which we denote as $\mathcal{N}_{ori,i}^t$ and $\mathcal{N}_{new,i}^t$ respectively. The ratio between $\mathcal{N}_{ori,i}^t$ and $\mathcal{N}_{new,i}^t$ represents how the N-gram's occurrence frequency changes when adding *new context*, which we denote as new occurrence ratio:

$$\Delta\mathcal{N}_K^{2K}(g_i^t) = \frac{\mathcal{N}_{new,i}^t + 1}{\mathcal{N}_{ori,i}^t + 1}, i \in [2K-1, |D_t|) \tag{7}$$

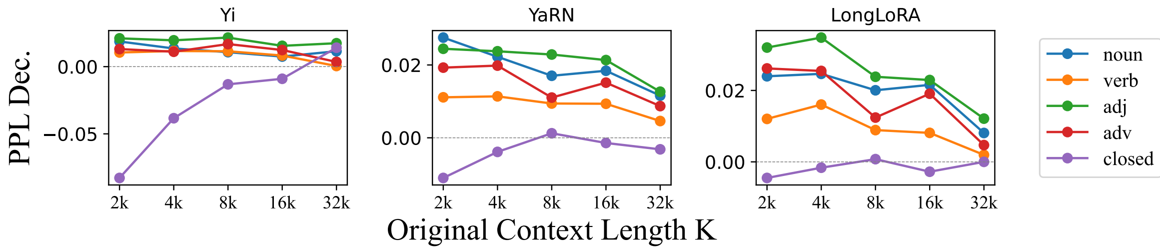

Figure 3: ΔD of each class of POS tags.

Then we calculate the average token-perplexity decrement $\Delta\bar{p}_K^{2K}$ and the average N-gram's new occurrence ratio $\Delta\mathcal{N}_K^{2K}$ over all documents:

$$\Delta\bar{p}_K^{2K} = \frac{1}{M}\sum_{t=1}^{M}\left(\frac{1}{|D_t|}\sum_{i=2K-1}^{|D_t|-1}\bar{p}_K^{2K}(x_i^t)\right) \quad (8)$$

$$\Delta\mathcal{N}_K^{2K} = \frac{1}{M}\sum_{t=1}^{M}\left(\frac{1}{|D_t|}\sum_{i=2K-1}^{|D_t|-1}\mathcal{N}_K^{2K}(g_i^t)\right) \quad (9)$$

which we abbreviate as $\Delta\bar{p}$ and $\Delta\mathcal{N}$ for brevity.

To figure out the relationship between the token-perplexity decrement $\Delta\bar{p}$ and the N-gram's new occurrence ratio $\Delta\mathcal{N}$, we adopt a widely used metric, Spearman's rank correlation coefficient, for analysis. Specifically, Spearman's rank correlation coefficient is computed over $\Delta\bar{p}(x_i^t)$ and $\Delta\mathcal{N}(g_i^t)$ of all tokens x_i^t in the test corpus. Here we show the result of N=5 and K ranges from 2k to 32k, and we will discuss the effect of N later.

As the results shown in Table 3, there is a strong correlation coefficient in every experimental setting. For example, with $K = 32k$, Yi delivers a correlation coefficient of 0.356. When $K = 2k$, the correlation coefficient is even higher, up to 0.53, demonstrating the positive correlation between the token-perplexity decrement $\Delta\bar{p}$ and N-gram's new occurrence ratio $\Delta\mathcal{N}$. The larger the new occurrence ratio of a token's N-gram, the more its token-perplexity decreases. Therefore, tokens with a higher frequency of N-gram in *new context* can benefit more from the additional long text.

Effect of N. We further analyze the effect of N on the correlation between $\Delta\bar{p}$ and $\Delta\mathcal{N}$. We fix K to 32k and calculate the Spearman's rank correlation coefficient with N ranging from 3 to 20, as shown in Figure 4. Circle markers represent correlation coefficients with p-value < 0.005 while x markers represent p-value $\geq$ 0.005, which indicates there is no correlation.

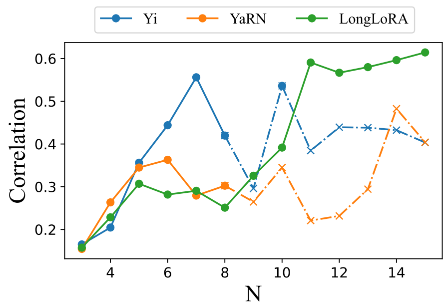

Figure 4: Correlation coefficients between the token-perplexity decrement $\Delta\bar{p}$ and the N-gram's new occurrence ratio $\Delta\mathcal{N}$ under different values of N.

There are always correlations between $\Delta\bar{p}$ and $\Delta\mathcal{N}$ when N ranges from 4 to 8. In LongLoRA, we can see the correlation become stronger with the increase of N, indicating under the same $\Delta\mathcal{N}$, a longer N-gram may more easily affect its corresponding token's prediction. Considering the impressive in-context learning ability (Brown et al., 2020) of LLMs, it may be one possible explanation for the strong correlation between $\Delta\bar{p}$ and $\Delta\mathcal{N}$ that LLMs tend to learn the N-grams frequently appearing in the input text.

5.3 Priors in Long-context LLMs

Priors in long-context LLMs may affect their performance in language modeling. For example, as pre-training is a crucial part of LLM training, the composition of pre-training data can greatly affect the LLMs' performance in downstream tasks. So, in the language modeling task, is the decrease in token-perplexity also affected by pre-training data? Specifically, if a token appears more frequently in the pre-training data and the LLM is fully familiar with this token, will the LLM be more sensitive, i.e., less affected by changes in the context, when predicting this token? From this perspective, we will explore the relationship between the frequency of tokens appearing in pre-training data and the

$\Delta\mathcal{N}$	$(0,1)$	$\{1\}$	$(1,\infty)$
Yi	0.049	0.463*	0.086
YaRN	0.014	0.330*	0.107
LongLoRA	0.047	0.385*	0.111

Table 4: Correlation coefficients between the token-perplexity decrement $\Delta\bar{p}$ and the token frequencies fr under different N-gram's new occurrence ratio $\Delta\mathcal{N}$. K=32k, N=5. *: p-value < 0.005.

change in token-perplexity.

Token's Frequency Calculation. As the data used to pre-train LLMs are rarely disclosed, we use RedPajama (Computer, 2023), a fully open-source reproduction of LLaMA (Touvron et al., 2023), as a proxy for calculating the tokens' frequency in LLMs' pretraining dataset. We randomly sample 20G tokens from RedPajama, where the proportion of each subset is determined according to the pretraining data sampling proportion mentioned in LLaMA. We calculate the frequency of each token $fr(x_i)$ in the sampled dataset to approximate the token frequency in the pre-training data.

Correlation between fr and $\Delta\bar{p}$ under Different $\Delta\mathcal{N}$. To evaluate the relationship between the token frequency and its token-perplexity change, we calculate the correlation between the token frequency and the degree of changes in token-perplexity. Note that, unlike the token-perplexity decrement $\Delta\bar{p}$ in the previous experiments, this experiment examines the degree of token-perplexity **changes** affected by the token frequency, which is calculated as:

$$\delta\bar{p}_K^{2K}(x_i) = abs(\bar{p}_{2K}(x_i) - \bar{p}_K(x_i)) \quad (10)$$

which we denoted as $\delta\bar{p}(x_i)$ for brevity.

Considering that the new occurrence ratio of a token's n-gram $\Delta\mathcal{N}(g_i^t)$ will also affect its token-perplexity change $\delta\bar{p}(x_i)$, we classified tokens into 4 groups based on $\Delta\mathcal{N}(g_i^t)$. This guarantees that the $\Delta\mathcal{N}(g_i^t)$ of the token x_i within the same group are close to each other, thus mitigating the impact of context information on token-perplexity changes. We group the tokens according to the following rules: Group A: $\{x_i|\Delta\mathcal{N} < 1\}$, Group B: $\{\Delta\mathcal{N} = 1\}$, Group C: $\{x_i|\Delta\mathcal{N} > 1\}$. For each group, we calculate the correlation coefficient between $\delta\bar{p}$ and fr.

The results are shown in Table 4. Only when $\Delta\mathcal{N} = 1$, i.e., the frequency of N-gram g_i does not change when adding new context, there is a strong correlation between $fr(x_i)$ and $\delta\bar{p}(x_i)$, such as correlation coefficient of 0.463 in Yi. In other cases, there is no correlation as their p-value > 0.005. The results indicate that the context information dominates the influence on the token-perplexity changes than tokens' frequency in the pretraining dataset. The long-context LLMs are more likely to use what they learned during the pretraining stage to predict the current token x_i when the N-gram g_i's frequency does not change in the context.

6 Why Perplexity Decreases

We also find that, apart from the tokens aforementioned benefit from the distant text, tokens that are incorrectly predicted by LLMs also show a higher average probability. So we wonder, whether long-context LLMs exhibit overconfidence when the context length increases. Specifically, we explore the probability distribution $\mathbb{P}$ output by the model when predicting each token. If the model becomes more confident, no matter whether or not the model correctly predicts the token x_i, the probability distribution $\mathbb{P}_{K,i}$ will become sharper, i.e., the entropy of $\mathbb{P}_{K,i}$ decreases and the $max(\mathbb{P}_{K,i})$ increases.

Therefore, we calculate the entropy $E_{K,i} = entropy(\mathbb{P}_{K,i})$ and the max probability $MP_{K,i} = max(\mathbb{P}_{K,i})$ of each token's probability distribution $\mathbb{P}_{K,i}$. Specifically, we split the tokens in the whole test corpus into two groups, **T** and **F**, based on whether the tokens are correctly predicted (**T**) by the model or not (**F**). Given context length K, for each group, we calculate the average entropy E_K and the average max probability MP_K of all tokens in the group following the equation:

$$E_K^{\mathbf{T}} = \frac{1}{|\mathbf{T}|}\sum_{i\in\mathbf{T}} E_{K,i}^t \quad (11)$$

$$MP_K^{\mathbf{T}} = \frac{1}{|\mathbf{T}|}\sum_{i\in\mathbf{T}} MP_{K,i}^t \quad (12)$$

Here we use group **T** as an example, $E_K^{\mathbf{F}}$ and $MP_K^{\mathbf{F}}$ of group **F** are calculated in the same way.

Figure 5 shows that, as the context length K increases, there are consistent trends between group **T** and **F** in both E_K and MP_K. Especially in group **F**, where LLMs make incorrect predictions, $E_K^{\mathbf{F}}$ decreases and $MP_K^{\mathbf{F}}$ increases, indicating the longer inputs lead to sharper probability distributions. Such a phenomenon shows that long-context LLMs are more confident with longer inputs.

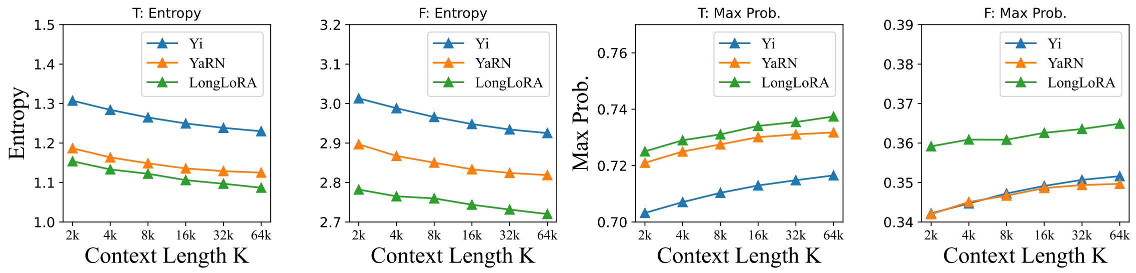

Figure 5: The entropy E_K and the max probability MP of groups **T** and **F** respectively.

Therefore, for group **T**, the $MP_K^{\mathbf{T}}$'s increase may be partly due to the sharper probability distributions from the more confident long-context LLMs. Note that, all tokens in **T** satisfy $argmax(\mathbb{P}_{K,i}^{\mathbf{T}}) = i$, i.e., $MP_{K,i} = p_K(x_i)$. According to the definition of perplexity in Equation 3, an increase in $MP_{K,i}$ will lead to a decrease in token-perplexity, which demonstrates that the increasing confidence of the long-context LLMs may be one of the reasons for the perplexity decrease.

7 Related Work

Long-Context LLMs. Extensive studies have aimed to scale up the context window of Large Language Models (LLMs) to handle long-context inputs (OpenAI, 2023; Anthropic, 2023; Chen et al., 2023b,a; Xiong et al., 2023; Ding et al., 2023a; Chen et al., 2023c). For example, GPT-4 (OpenAI, 2023) has a 128k context window, and Yi (AI et al., 2024) supports a context window of 200k.

Some long-context LLMs use the length extrapolation approach in Transformers (Vaswani et al., 2017), which is trained on short sequences while inferring on long sequences, to handle long text (Press et al., 2022; Sun et al., 2023; Su et al., 2023b). While some other research, such as Position Interpolation (Chen et al., 2023b), NTK-aware position embeddings (bloc97, 2023), and YaRN (Peng et al., 2023a), propose positional interpolation methods for long text processing.

Besides, LLMs like LongLoRA (Chen et al., 2024) and LongNet (Ding et al., 2023a) focus on the efficient attention calculation. Retrieval-based approaches (Tworkowski et al., 2023; Wang et al., 2024; Borgeaud et al., 2022), recurrent transformers (Bulatov et al., 2022; Staroverov et al., 2024) and prompt compression (Jiang et al., 2023) are also effective methods for context window extension.

Long-context LLMs Evaluation. Multiple benchmarks have been proposed for long-context LLMs evaluation. ZeroSCROLLS (Shaham et al., 2023) is a zero-shot benchmark containing ten natural language tasks. L-Eval (An et al., 2024) encompasses 18 realistic natural language tasks, such as QA, summarization, and math. Similarly, LongBench (Bai et al., 2023) incorporates 21 tasks of four categories. InfiniteBench (Zhang et al., 2024) is proposed with average data length surpassing 100K tokens. RULER (Hsieh et al., 2024) proposes synthetic tasks of four categories, aiming to provide flexibility to control the context lengths and task complexities.

Apart from downstream tasks such as QA, summarization and retrieval, language modeling is also widely used for long-context LLMs evaluation (Chen et al., 2024; Peng et al., 2023a; AI et al., 2024; Ding et al., 2023b; Men et al., 2024), which use Perplexity (PPL) as the evaluation metric to access LLMs' long text language modeling ability.

8 Conclusion

Different from human habits, a great number of tokens can benefit from additional distant text in long-context language modeling of LLMs. Specifically, content words and the starting token of a word benefit most from the long text. Patterns' frequency (N-gram) also plays an important role in token predictions. Besides, tokens of high frequency in the pre-training dataset show less sensitivity to the extension of the text. Furthermore, we observe that the overconfidence of long-context LLMs when the context length increases may be one possible reason for the perplexity decrease. We hope our analysis can provide insights for a better understanding of long-context LLMs and help the community design more reliable long-context LLMs.

Limitations

Due to the computational resources limitation, we can only conduct experiments with context length K ranging from 2k to 64k. We get out-of-memory when K=128k. Besides, it is worthy to further investigate why some tokens continuously benefit from the additional text even if the text is extremely far away. We will leave it as our future work.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Chenxin An, Shansan Gong, Ming Zhong, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024. L-eval: Instituting standardized evaluation for long context language models. In *ICLR*.

Anthropic. 2023. Model card and evaluations for claude models.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *Preprint*, arXiv:2308.14508.

Alan Bell, Jason M. Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2009. Predictability effects on durations of content and function words in conversational english. *Journal of Memory and Language*, 60(1):92–111.

bloc97. 2023. NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. *Preprint*, arXiv:2112.04426.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. 2022. Recurrent memory transformer. *Preprint*, arXiv:2207.06881.

Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. 2023a. Clex: Continuous length extrapolation for large language models. *Preprint*, arXiv:2310.16450.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023b. Extending context window of large language models via positional interpolation. *Preprint*, arXiv:2306.15595.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023c. Longlora: Efficient fine-tuning of long-context large language models. *Preprint*, arXiv:2309.12307.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024. LongLoRA: Efficient fine-tuning of long-context large language models. In *ICLR*.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Nelson Cowan. 2010. The magical mystery four: How is working memory capacity limited, and why? *Curr Dir Psychol Sci. 2010;19(1):51-57. doi:10.1177/0963721409359277*.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. 2023a. Longnet: Scaling transformers to 1,000,000,000 tokens. *Preprint*, arXiv:2307.02486.

Jiayu Ding et al. 2023b. LongNet: Scaling Transformers to 1,000,000,000 tokens. *arXiv:2307.02486*.

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

Huiqiang Jiang et al. 2023. LongLlmLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. *arXiv:2310.06839*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Xin Men, Mingyu Xu, Bingning Wang, Qingyu Zhang, Hongyu Lin, Xianpei Han, and Weipeng Chen. 2024. Base of rope bounds context length. *Preprint*, arXiv:2405.14591.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023a. Yarn: Efficient context window extension of large language models. *Preprint*, arXiv:2309.00071.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023b. Yarn: Efficient context window extension of large language models. *Preprint*, arXiv:2309.00071.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *Preprint*, arXiv:1911.05507.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *EMNLP*.

Alexey Staroverov, Egor Cherepanov, Dmitry Yudin, Alexey K. Kovalev, and Aleksandr I. Panov. 2024. Recurrent action transformer with memory. *Preprint*, arXiv:2306.09459.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2023a. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, page 127063.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023b. RoFormer: Enhanced Transformer with rotary position embedding. *arXiv:2104.09864*.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. 2023. A length-extrapolatable Transformer. In *Proc. of the 61st Annual Meeting of the ACL (Volume 1: Long Papers)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2023. Focused transformer: Contrastive training for context scaling. *Preprint*, arXiv:2307.03170.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2024. Augmenting language models with long-term memory. *NeurIPS*, 36.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. 2023. Effective long-context scaling of foundation models. *Preprint*, arXiv:2309.16039.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *Preprint*, arXiv:2201.05337.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞bench: Extending long context evaluation beyond 100k tokens. *arXiv:2402.13718*.

Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2023. Pose: Efficient context window extension of llms via positional skip-wise training. *ArXiv*, abs/2309.10400.

A Sliding Window Evaluation of Perplexity

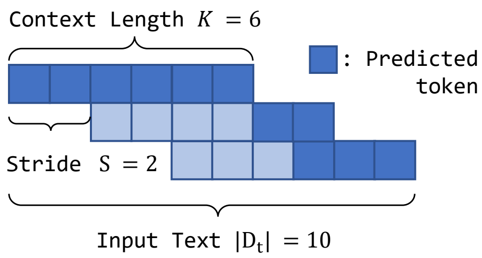

Figure 6: An illustration for sliding window method with stride S=2.

K	2k	4k	8k	16k	32k	64k
S	10	25	50	100	200	400

Table 5: Context length K and its corresponding stride S.

Here, we will briefly introduce the sliding window evaluation (Press et al., 2021) of perplexity in evaluating long-context LLMs. Given M documents $\{D_1, ..., D_M\}$, each D_t is split into chunks with stride S and the length of each chunk is K, which is also denoted as **context length** K. In each chunk, long-context LLMs predict token x_i based on previous tokens in the chunk C_q, and output the probability of token x_i:

$$p_K(x^t_{q,i}) = P(x^t_{q,i}|x^t_{q,0}, ..., x^t_{q,i-1}), i \in [0, K) \tag{13}$$

As illustrated in Figure 6, we obtain the probabilities of all tokens in the first chunk. Meanwhile, for other chunks, we only obtain the probability of the last S tokens. This way, we can finally get all tokens' probabilities in D_t. Then perplexity (PPL) is calculated as Equation 4. The calculation method we mentioned in Section 2 is the case where S=1.

Note that, except for the first chunk, probabilities of tokens in other chunks are only recorded when $i \in [K - S, K)$. For fair comparison among all tokens' token-perplexity, we need to ensure every token is predicted based on a similar length of input text. Therefore, in our experiment, we set $K >> S$ to ensure all tokens are predicted by long-context LLMs based on nearly the same number of previous tokens. The values of K and S are shown in Table 5.