Analysing The Impact of Sequence Composition on Language Model Pre-Training

Yu Zhao [†✉] **Yuanbin Qu** [‡] **Konrad Staniszewski** [§] **Szymon Tworkowski** [§]
Wei Liu [‡] **Piotr Miłoś** [§] **Yuxiang Wu** [¶] **Pasquale Minervini** [†✉]
[†] University of Edinburgh [‡] Xiaomi AI Lab [§] University of Warsaw [¶] Weco AI
yu.zhao@ed.ac.uk p.minervini@ed.ac.uk

Abstract

Most language model pre-training frameworks concatenate multiple documents into fixed-length sequences and use *causal masking* to compute the likelihood of each token given its context; this strategy is widely adopted due to its simplicity and efficiency. However, to this day, the influence of the pre-training sequence composition strategy on the generalisation properties of the model remains under-explored. In this work, we find that applying causal masking can lead to the inclusion of distracting information from previous documents during pre-training, which negatively impacts the performance of the models on language modelling and downstream tasks. In *intra-document causal masking*, the likelihood of each token is only conditioned on the previous tokens in the same document, eliminating potential distracting information from previous documents and significantly improving performance. Furthermore, we find that concatenating related documents can reduce some potential distractions during pre-training, and our proposed efficient retrieval-based sequence construction method, BM25Chunk, can improve in-context learning (+11.6%), knowledge memorisation (+9.8%), and context utilisation (+7.2%) abilities of language models without sacrificing efficiency.

1 Introduction

Large Language Models (LLMs) are pre-trained on large amounts of documents by optimising a language modelling objective and show an intriguing ability to solve a variety of downstream NLP tasks (Brown et al., 2020; Biderman et al., 2023; Touvron et al., 2023; Jiang et al., 2023). Previous works emphasise the importance of pre-training data quality (e.g., Gunasekar et al., 2023; Lee et al., 2022; Tirumala et al., 2023; Soboleva et al., 2023) and diversity (e.g., Xie et al., 2023; Gao et al., 2021; Kaddour, 2023) to improve the generalisation properties of language models. However, the influence of the pre-training sequence composition strategy remains largely under-explored.

For most decoder-only language model pre-training pipelines (e.g., Shoeybi et al., 2019; Ott et al., 2019; Brown et al., 2020; Biderman et al., 2023; Geng, 2023; Liu et al., 2023b; Zhang et al., 2024), constructing a pre-training instance involves *packing*, which refers to the process of combining randomly sampled documents into a *chunk* that matches the size of the context window; and *causal masking*, which refers to predicting the next token conditioned on all previous tokens, including those from different documents in the chunk. An alternative to causal masking is *intra-document causal masking*, where the likelihood of each token is conditioned on the previous tokens from the same document; intra-document causal masking is not commonly used in existing open-source pre-training frameworks as it is argued to adversely impact pre-training efficiency (Brown et al., 2020; Pagliardini et al., 2023). However, to the best of our knowledge, there is no systematic analysis in the literature on how causal masking affects the generalisation properties of models despite its role in improving efficiency.

To analyse the impact of the packing and masking strategies on pre-training, we pre-train language models using intra-document causal masking (referred to as INTRADOC, Section 2.2) and compare them with models pre-trained via causal masking with several *packing* strategies by varying the relatedness of the documents in the pre-training chunks. Specifically, we analyse the results produced by a commonly used baseline method that randomly samples and packs documents (MIXCHUNK); a method that samples and packs documents from the same source based on their meta-information

https://github.com/yuzhaouoe/pretraining-data-packing

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7897–7912
August 11-16, 2024

(UNIChunk); and our proposed efficient retrieval-based packing method, which retrieves and packs related documents (BM25Chunk, Section 2.1).

Our experimental results indicate that using causal masking without considering the boundaries of documents can lead to the inclusion of distracting information from previous documents during pre-training (Section 3 and Section 5.1), negatively impacting the performance of the models in downstream tasks (Section 4). We observe that intra-document causal masking, which eliminates the potential distractions from irrelevant documents during pre-training, can significantly improve the performance of the model while increasing its runtime (+4% in our implementation, see Appendix A).

We also find that improving the relatedness of the documents in pre-training chunks can reduce some potential distractions from previous documents (e.g., UNIChunk avoids packing documents from different distributions, such as code and news text), which can improve the performance of causal masking models on a wide array of downstream tasks. Finally, we show that our proposed efficient retrieval-based packing method, BM25Chunk, can improve a model's language modelling (+6.8%), in-context learning (+11.6%), knowledge memorisation (+9.8%), and context utilisation (+7.2%) abilities using causal masking and thus without sacrificing pre-training efficiency.

Our main contributions and findings can be summarised as follows:

- We systematically analyse and compare the models pre-trained using causal masking and intra-document causal masking; our experimental results reveal that using causal masking without considering the boundaries of documents can result in significant performance degradation (Section 3 and Section 4).
- We find that improving the relatedness of the documents in each pre-training chunk benefits causal masking models, and our proposed efficient retrieval-based packing method (BM25Chunk, Section 2.1) can improve the performance of language models significantly.
- We quantitatively analyse the attention distribution of the models during language modelling (Section 5.1), and investigate the burstiness property of pre-training chunks (Section 5.2); our findings indicate that models can be more robust to irrelevant contexts and obtain better performance when improving the relatedness of documents in pre-training chunks.

2 Packing and Masking Strategies for Pre-Training Sequence Composition

In this section, we formally introduce the pre-training data packing strategies, causal masking, and intra-document causal masking.

2.1 Packing Strategies

Let $\mathcal{D}_i$ represent a corpus, such as Wikipedia, C4, or GitHub, and let $\mathcal{D} = \bigcup_s \mathcal{D}_s$ denote the dataset resulting from the union of such corpora. Furthermore, each corpus $\mathcal{D}_s$ is defined as a set of documents $\mathcal{D}_s = \{d_1, \ldots, d_{|\mathcal{D}_s|}\}$, where each document d_i is defined as a sequence of tokens $d_i = (x_1, \ldots, x_{|d_i|})$.

A *packing strategy* involves first selecting a set of documents $\{d_i\}_{i=1}^n$ from $\mathcal{D}$, and then packing them into a chunk C with a fixed length $|C| = L$. Following Brown et al. (2020), we concatenate the documents $\{d_i\}_{i=1}^n$ by interleaving them with end-of-sentence ([EOS]) tokens to construct a chunk. A *packed sequence* (or *chunk*) C is denoted as:

$$C = (d_1[\text{EOS}]d_2[\text{EOS}]\ldots\text{SPLIT}(d_n)), \quad (1)$$

where $[\text{EOS}]$ is the end-of-sentence token, $\text{SPLIT}()$ truncates the last document such that $|C| = L$, and the content of the chunk C will be removed from the dataset $\mathcal{D}$ to avoid sampling the same documents multiple times.

In the following, we introduce three strategies to sample the documents $\{d_i\}_{i=1}^n$ from the dataset $\mathcal{D}$ for composing each pre-training chunk, namely MIXChunk, UNIChunk, and BM25Chunk.

MIXChunk In MIXChunk (baseline), documents $d_i \in \mathcal{D}$ are sampled uniformly at random from the entire pre-training corpus $\mathcal{D}$:

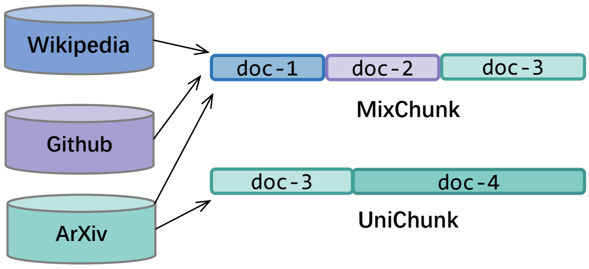
$$d_i \sim \text{Uniform}(\mathcal{D}).$$

As a result, in MIXChunk, a chunk can contain documents from different source datasets, *e.g.*, Wikipedia and GitHub, as shown in Figure 1(a).

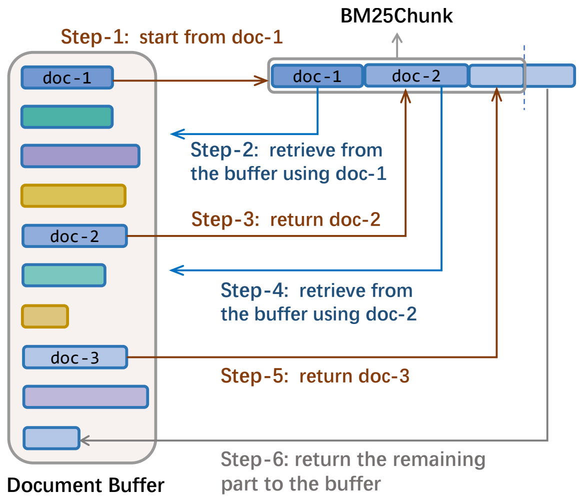
UNIChunk In UNIChunk, each chunk is composed of documents from a single source corpus $\mathcal{D}_s$:

$$d_i \sim \text{Uniform}(\mathcal{D}_s), \quad \text{with } \mathcal{D}_s \subseteq \mathcal{D}.$$

This helps to avoid packing documents from different distributions (such as code and news text) together. To construct a training batch, we sample sequences from each corpus $\mathcal{D}_s$ in proportion to the number of tokens in $\mathcal{D}_s$.



(a) MIXCHUNK randomly samples documents from all corpora to construct pre-training sequences, which can pack documents from different sources. UNICHUNK randomly samples documents from a single source to construct a sequence.



(b) The sequence construction process in BM25CHUNK. The left part represents a document buffer that caches a set of documents randomly sampled from the corpus.

Figure 1: Packing strategies for pre-training chunks construction. (a) illustrates the compositions of MIXCHUNK and UNICHUNK; (b) presents the sequence construction process of BM25CHUNK.

BM25CHUNK To improve the relevance of documents in pre-training chunks, we employ a BM25-based retriever to construct pre-training chunks, referred to as BM25CHUNK. Specifically, given a document $d_i \in \mathcal{D}_s$, we retrieve a sequence of documents $\{d_i\}_{i=1}^n$ by $d_{i+1} = \textsc{Retrieve}(d_i, \mathcal{D}_s)$; here, $\textsc{Retrieve}(d_i, \mathcal{D}_s)$ retrieves the most similar documents to d_i from $\mathcal{D}_s$ based on BM25 scoring.

However, this retrieval process can be computationally inefficient due to the size of the pre-training corpus $\mathcal{D}_s$. To improve the efficiency of the retrieval step, we restrict the retrieval scope to a subset $\mathcal{B}_s \subseteq \mathcal{D}_s$ of the corpus $\mathcal{D}_s$, reducing the computational complexity of retrieval; the proposed approach is outlined in Figure 1(b). More formally, we introduce a document buffer $\mathcal{B}_s \subseteq \mathcal{D}_s$ that contains k documents uniformly sampled from $\mathcal{D}_s$, which serves as the retrieval source for constructing pre-training chunks:

$$d_1 \sim \text{Uniform}(\mathcal{B}_s), \quad d_{i+1} = \textsc{Retrieve}(d_i, \mathcal{B}_s).$$

After retrieving a sequence of documents $\{d_i\}_{i=1}^n$ from the buffer $\mathcal{B}_s$ for constructing a chunk, we refill the buffer by sampling new documents from $\mathcal{D}_s$. The time complexity analysis and more details are presented in Appendix C.

2.2 Masking Strategies

Another core element of LLM pre-training is the *masking* strategy, which determines how next-token prediction distributions are conditioned on other tokens in the sequence.

Causal Masking In causal masking, each token in a sequence is predicted solely based on all preceding tokens in the sequence. More formally, given a chunk $C = (x_1, \ldots, x_{|C|})$ defined as in Equation (1), the likelihood of C is given by:

$$P(C) = \prod_{i=1}^{|C|} P(x_i \mid x_1, \ldots, x_{i-1}),$$

where $P(x_i \mid x_1, \ldots, x_{i-1})$ denotes the probability of the token x_i given all preceding tokens $x_1, \ldots, x_{i-1}$ in the chunk. During pre-training, *causal masking* implies that, given a chunk C, the probability of each token in C will be conditioned on all preceding tokens, including those belonging to different documents. Causal masking is the *standard practice* when pre-training decoder-only LLMs (e.g., Shoeybi et al., 2019; Brown et al., 2020; Zhang et al., 2022; Biderman et al., 2023; Geng, 2023; Liu et al., 2023b; Zhang et al., 2024).

Intra-Document Causal Masking In intra-document causal masking, on the other hand, the probability of each token is conditioned on the previous tokens within the same document. More formally, given a chunk C defined as in Equation (1), the probability of each token d_{ij} belonging to document d_i is only conditioned on the preceding tokens within d_i:

$$P(C) = \prod_{i=1}^{n} \prod_{j}^{|d_i|} P\left(d_{ij} \mid d_{i1}, \ldots, d_{i(j-1)}\right).$$

We refer to models trained using intra-document causal masking as INTRADOC. The details of our efficient implementation using FlashAttention (Dao, 2023) are available in Appendix A.

L	Model	CommonCrawl	C4	Wikipedia	GitHub	StackExchange	Book	ArXiv	Avg.
2K	MixChunk	13.284	13.884	6.811	5.531	8.051	11.623	5.203	9.172
	UniChunk	11.805	<u>13.650</u>	6.546	5.518	7.839	11.353	5.106	$8.831_{\downarrow 0.341}$
	Bm25Chunk	**11.418**	13.677	<u>6.237</u>	<u>4.585</u>	<u>7.623</u>	<u>11.253</u>	<u>5.059</u>	<u>8.550</u>$_{\downarrow 0.622}$
	IntraDoc	<u>11.631</u>	**13.197**	**6.084**	**4.252**	**7.535**	**11.130**	**5.030**	**8.410**$_{\downarrow 0.883}$
8K	MixChunk	9.645	14.424	7.010	7.496	8.634	11.337	4.911	9.065
	UniChunk	9.478	14.190	6.897	7.006	8.456	11.117	4.812	$8.851_{\downarrow 0.214}$
	Bm25Chunk	<u>9.144</u>	<u>13.579</u>	<u>6.287</u>	<u>5.463</u>	<u>8.022</u>	<u>10.810</u>	<u>4.715</u>	<u>8.289</u>$_{\downarrow 0.776}$
	IntraDoc	**8.994**	**13.173**	**6.073**	**5.010**	**7.894**	**10.701**	**4.705**	**8.079**$_{\downarrow 0.986}$

Table 1: Evaluation of perplexity on SlimPajama's test set. The best score is highlighted in bold, and the second best is highlighted with an underline. L is the maximum length of the sequence for pre-training. Subscript $_\downarrow$ presents the PPL improvement over the *baseline* method MixChunk. We report the results of next-token accuracy in Appendix F.

3 Language Model Pre-Training

3.1 Settings

Pre-Training Corpora In this work, we use SlimPajama (Soboleva et al., 2023) as the pre-training corpus, which consists of seven sub-corpora, including CommonCrawl, C4, Wikipedia, GitHub, StackExchange, ArXiv, and Book. This allows us to investigate packing strategies in a mixed corpora setting. We sample documents with 150B tokens from SlimPajama as the pre-training corpus and ensure each subset maintains the same proportion of tokens as in the original dataset.

Pre-Training Models The model implementation is based on the LLaMA (Touvron et al., 2023) architecture with minor modifications to support intra-document causal masking. We pre-train 1.3B parameters models using context windows of 2,048 (referred to as 2K) and 8,192 (8K) tokens. We use the same set of documents with the difference in pre-training sequence composition to pre-train models, including causal masking models, i.e., MixChunk, UniChunk, and Bm25Chunk, and intra-document causal masking models IntraDoc. More details are available in Appendix B.

Previous works (Brown et al., 2020; Pagliardini et al., 2023) argued that dynamic sequence-specific sparse masking reduces training efficiency. Compared to causal masking, we observe a 4.0% efficiency degradation on intra-document causal masking in our implementation, and the discussion on implementation is presented in Appendix A.

3.2 Results

For evaluating LLMs trained under different packing strategies, in this work, we compute the perplexity (PPL) of a held-out set of documents where each document is treated independently. The results are summarised in Table 1.

We can see that Bm25Chunk achieves the lowest PPL among the three causal masking models, yielding a lower average PPL compared to MixChunk in the 2K (-0.62) and 8K (-0.78) settings. Furthermore, UniChunk also yields a lower average PPL than the baseline MixChunk (-0.34 and -0.21). These results indicate that increasing the relatedness of documents in a sequence can improve the language modelling ability of models.

When considering models trained via intra-document causal masking, we can see that IntraDoc achieves the lowest PPL compared to all models trained via causal masking. This indicates eliminating the potential distracting information from irrelevant documents during pre-training benefits the language modelling ability of models. Specifically, we observe that both Bm25Chunk and IntraDoc obtain significantly lower PPLs on GitHub, where IntraDoc improves over UniChunk in both the 2K (-1.3 PPL) and 8K (-2.0) models. For UniChunk, though we avoided packing web text and code, its improvement over MixChunk on GitHub is slight. This phenomenon could imply that *code pre-training is more adversely affected by the distraction of unrelated context*, and both intra-document causal masking and retrieval-based sequence construction strategy can alleviate this issue.

4 Experiments on Downstream Tasks

In the following, we evaluate the in-context learning, knowledge memorisation, and context utilisation abilities of the models.

4.1 In-Context Learning

Following Shi et al. (2023), we evaluate in-context learning abilities of the models using seven text classification datasets, namely SST2 (Socher et al., 2013), Amazon (Zhang et al., 2015), Yelp (Zhang

L	Model	SST2	Amazon	DBpedia	AGNews	Yelp	Hate	Offensive	Avg.
2K	MixChunk	$71.53_{\pm 13.8}$	$81.57_{\pm 15.7}$	$40.87_{\pm 3.34}$	$74.98_{\pm 0.99}$	$86.89_{\pm 4.81}$	$47.10_{\pm 7.51}$	$41.82_{\pm 20.46}$	63.54
	UniChunk	$77.61_{\pm 10.05}$	$90.88_{\pm 1.13}$	$36.61_{\pm 2.15}$	$70.39_{\pm 2.23}$	$91.16_{\pm 0.35}$	$46.20_{\pm 5.67}$	$42.30_{\pm 14.92}$	65.02
	Bm25Chunk	$\mathbf{83.73}_{\pm 8.17}$	$\mathbf{90.90}_{\pm 3.20}$	$\mathbf{50.16}_{\pm 2.61}$	$\mathbf{75.98}_{\pm 2.73}$	$91.67_{\pm 3.68}$	$48.58_{\pm 5.26}$	$55.36_{\pm 15.10}$	**70.91**
	IntraDoc	$73.65_{\pm 13.61}$	$84.06_{\pm 12.68}$	$46.82_{\pm 1.82}$	$72.32_{\pm 2.66}$	$\mathbf{91.91}_{\pm 0.97}$	$\mathbf{55.72}_{\pm 3.47}$	$\mathbf{69.14}_{\pm 5.37}$	70.52
8K	MixChunk	$76.01_{\pm 8.14}$	$87.32_{\pm 3.08}$	$45.94_{\pm 3.70}$	$68.21_{\pm 6.21}$	$79.06_{\pm 9.99}$	$42.85_{\pm 1.19}$	$37.03_{\pm 14.28}$	62.43
	UniChunk	$\mathbf{81.61}_{\pm 8.63}$	$88.30_{\pm 2.68}$	$52.84_{\pm 2.36}$	$63.16_{\pm 9.25}$	$83.45_{\pm 6.41}$	$45.50_{\pm 3.00}$	$46.84_{\pm 16.78}$	65.96
	Bm25Chunk	$80.87_{\pm 6.16}$	$91.39_{\pm 1.30}$	$56.57_{\pm 2.33}$	$\mathbf{74.79}_{\pm 2.89}$	$85.19_{\pm 6.93}$	$\mathbf{49.12}_{\pm 5.17}$	$48.33_{\pm 15.88}$	69.47
	IntraDoc	$72.38_{\pm 3.97}$	$\mathbf{93.25}_{\pm 0.91}$	$\mathbf{61.85}_{\pm 6.89}$	$72.49_{\pm 4.72}$	$\mathbf{92.83}_{\pm 1.38}$	$46.20_{\pm 3.26}$	$\mathbf{59.59}_{\pm 9.88}$	**71.23**

Table 2: In-context learning performance evaluated by text classification accuracy across seven datasets. Accuracy and deviation (subscript) are calculated using different sets of demonstrations sampled by 16 random seeds.

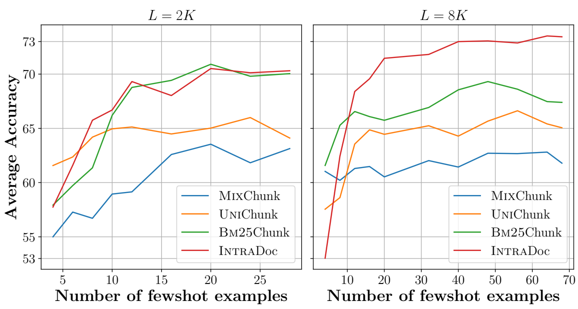


Figure 2: Average in-context learning accuracy using different numbers of few-shot demonstrations – the left and right figures show the results of 2K and 8K models.

L	Model	NQ	TQA	Avg.
2K	MixChunk	$6.19_{\pm 0.24}$	$14.47_{\pm 0.75}$	10.33
	UniChunk	$6.70_{\pm 0.26}$	$15.53_{\pm 0.74}$	11.12
	Bm25Chunk	$7.10_{\pm 0.27}$	$15.57_{\pm 0.65}$	11.34
	IntraDoc	$\mathbf{7.17}_{\pm 0.33}$	$\mathbf{16.04}_{\pm 0.35}$	**11.60**
8K	MixChunk	$5.08_{\pm 0.14}$	$10.90_{\pm 1.34}$	7.99
	UniChunk	$5.25_{\pm 0.37}$	$10.59_{\pm 1.10}$	7.92
	Bm25Chunk	$5.37_{\pm 0.43}$	$11.09_{\pm 0.67}$	8.23
	IntraDoc	$\mathbf{6.89}_{\pm 0.08}$	$\mathbf{15.09}_{\pm 0.79}$	**10.99**

Table 3: Exact Match scores on closed-book closed-book QA tasks.

et al., 2015), DBpedia (Lehmann et al., 2015), AGNews (Zhang et al., 2015), and TweetEval hate/offensive tweet detection tasks (Barbieri et al., 2020).

In Table 2, we report the in-context learning accuracy values of the models in few-shots learning settings, using 20 and 48 demonstrations for 2K and 8K models, respectively. We truncate the input sequences to fit within their respective context windows. For models pre-trained using causal masking, we can see that UniChunk produces more accurate results than MixChunk, while Bm25Chunk yields a higher average accuracy than MixChunk for 2K (+11.6%) and 8K (+11.3%) models. These results indicate that *increasing relatedness of the documents in pre-training chunks can improve the in-context learning abilities of the models.*

In Figure 2, we present the average accuracy using different numbers of few-shot demonstrations. We observe that Bm25Chunk has an on-par accuracy with IntraDoc on the 2K setting; however, IntraDoc obtains a significantly higher accuracy compared to Bm25Chunk on the 8K setting. It may imply that using a longer context window size can result in increased distractions for causal masking pre-training; meanwhile, constrained by the performance of the retrieval method, Bm25Chunk decreases the accuracy on the 8K setting. For 8K models, MixChunk and UniChunk obtain similar results to their corresponding 2K models, and they do not improve the accuracy when increasing the number of demonstrations. It might be due to the similar levels of distraction in both 2K and 8K settings using random packing strategies.

4.2 Knowledge Memorisation

We use two open-domain question-answering (ODQA) datasets, namely NaturalQuestions (NQ, Kwiatkowski et al., 2019) and TriviaQA (TQA, Joshi et al., 2017), to evaluate the knowledge memorisation properties of the models. We use 12 demonstrations for the 2K models and 48 demonstrations for the 8K models. In Table 3, we show the mean Exact Match (EM) scores calculated based on 5 different sets of demonstrations.

For models trained with causal masking, we can see that *increasing the relatedness of documents in pre-training chunks can improve the knowledge memorisation ability of models*. Compared to the baseline MixChunk, Bm25Chunk obtains +9.8% and +3.0% EM improvements on 2K and 8K models, respectively. We also note that intra-document causal masking significantly improves the knowledge memorisation ability, especially for 8K models, where IntraDoc improves EM by +12.3%

L	Model	RACE-h	RACE-m	SQuAD	HotpotQA	NQ-open	TQA-open	Avg.
2K	MIXChunk	$32.34_{\pm 0.43}$	$42.77_{\pm 0.69}$	$36.70_{\pm 1.79}$	$7.32_{\pm 1.31}$	$20.00_{\pm 0.46}$	$42.72_{\pm 1.37}$	30.31
	UNIChunk	$\underline{34.01}_{\pm 0.52}$	$43.52_{\pm 0.44}$	$37.33_{\pm 2.31}$	$7.12_{\pm 1.35}$	$21.16_{\pm 0.96}$	$42.32_{\pm 1.10}$	30.91
	BM25Chunk	$33.17_{\pm 0.36}$	$\underline{44.92}_{\pm 0.46}$	$\underline{37.91}_{\pm 1.84}$	$\mathbf{10.30}_{\pm 0.42}$	$\mathbf{22.10}_{\pm 0.91}$	$\mathbf{46.24}_{\pm 0.63}$	<u>32.42</u>
	INTRADoc	$\mathbf{34.49}_{\pm 0.56}$	$\mathbf{44.96}_{\pm 0.59}$	$\mathbf{39.91}_{\pm 1.48}$	$\underline{8.29}_{\pm 1.27}$	$\underline{21.66}_{\pm 0.85}$	$\underline{45.67}_{\pm 1.02}$	**32.49**
8K	MIXChunk	$31.66_{\pm 0.47}$	$41.57_{\pm 0.44}$	$32.79_{\pm 1.56}$	$10.53_{\pm 0.70}$	$20.53_{\pm 0.58}$	$40.53_{\pm 1.03}$	29.60
	UNIChunk	$31.68_{\pm 0.94}$	$41.64_{\pm 0.55}$	$34.94_{\pm 1.84}$	$10.57_{\pm 1.13}$	$21.76_{\pm 0.80}$	$39.60_{\pm 1.77}$	30.03
	BM25Chunk	$\underline{32.63}_{\pm 0.68}$	$\underline{44.14}_{\pm 0.48}$	$\underline{39.45}_{\pm 1.05}$	$\mathbf{14.46}_{\pm 0.93}$	$\underline{22.17}_{\pm 1.02}$	$\underline{43.40}_{\pm 0.38}$	**34.54**
	INTRADoc	$\mathbf{33.17}_{\pm 0.37}$	$\mathbf{45.56}_{\pm 0.38}$	$\mathbf{41.32}_{\pm 2.28}$	$\underline{12.60}_{\pm 1.49}$	$\mathbf{22.25}_{\pm 0.13}$	$\mathbf{44.19}_{\pm 0.60}$	<u>33.18</u>

Table 4: Evaluation results of machine reading comprehension and retrieval-augmented generation tasks.

and +37.5% over MIXChunk for 2K and 8K models, respectively. These results support our hypothesis that reducing the distractions deriving from concatenating multiple, potentially unrelated documents in pre-training chunks can improve the knowledge memorisation ability of the models.

4.3 Reading Comprehension and Retrieval-Augmented Generation

We evaluate the pre-trained models on a set of reading comprehension tasks, namely RACE (Lai et al., 2017), SQuAD (Rajpurkar et al., 2016), HotpotQA (Yang et al., 2018), and the following retrieval-augmented generation (RAG) tasks: NQ, TQA, and Multi-Document Question-Answering (MDQA, Liu et al., 2023a). For NQ and TQA, we use the top two passages retrieved by the dense retriever (Karpukhin et al., 2020; Izacard and Grave, 2021), denoted as NQ-open and TQA-open. Our results for RACE, SQuAD, and RAG tasks are summarised in Table 4, while the results on MDQA are available in Figure 3.

We can see that BM25Chunk produces more accurate results than MIXChunk and UNIChunk in all tasks and obtains the best average accuracy, showing that *increasing the relatedness of documents in pre-training chunks can improve the context utilisation ability*. Specifically, BM25Chunk obtains a significantly better accuracy on multi-hop QA task HotpotQA, showing it can better utilise multiple relevant information from the context.

INTRADoc obtains the best average accuracy in the 2K models and obtains the best scores in 5 of 6 tasks in the 8K models. It indicates that eliminating potential distractions from unrelated documents and *learning each document independently can improve context utilisation ability*. This finding is different from the ideas in previous works, which suggested that pre-training with multiple documents in one context (Shi et al., 2023) and adding distraction in context during pre-training (Tworkowski et al., 2023) benefit context utilisation ability.

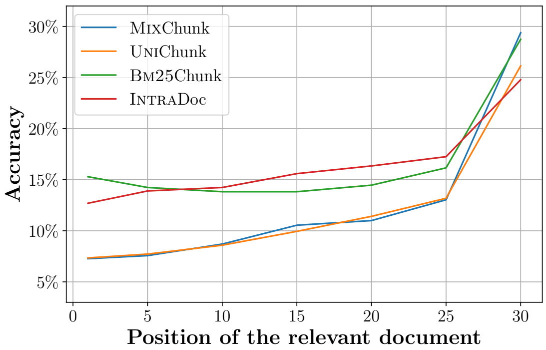
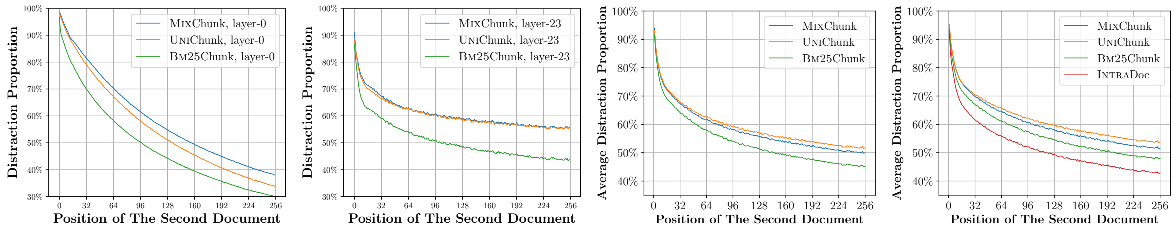


Figure 3: Accuracy on Multi-Document Question-Answering (MDQA). The x-axis represents the position of the document that contains the answer. The y-axis presents the accuracy for a position.

In MDQA, for each question, there are 30 documents provided in the context, where only one of them contains the answer to the question — MDQA is used to evaluate the ability of models to filter out irrelevant information and identify the relevant parts of a long context. This task has been used to analyse the *lost-in-the-middle* phenomenon in LLMs where they struggle to retrieve information stored in the middle of long contexts (Liu et al., 2023a). In the following, we analyse how the accuracy of models varies with the position of relevant information in the context. In these experiments, we focus on 8K models due to their ability to handle long contexts. The zero-shot results on MDQA are outlined in Figure 3. We observe that both BM25Chunk and INTRADoc tend to produce more accurate predictions than MIXChunk and UNIChunk when the relevant passage is located at the beginning or middle of the context. These results show that BM25Chunk and INTRADoc *can better filter irrelevant context and locate relevant information*; these results are further corroborated



(a) The distraction proportion of the *first layer*; different documents are separated by [EOS].

(b) The distraction proportion of the *last layer*; different documents are separated by [EOS].

(c) The average distraction proportion over layers; different documents are separated by [EOS].

(d) The average distraction proportion over layers; different documents are separated by "\n"

Figure 4: Distracted attention proportions of models. The x-axis presents the token position of the second document; the y-axis presents the distraction proportion calculated by Equation (2). Figures (a) and (b) show the distraction proportion of the first and last layers. Figures (c) and (d) are the average distraction proportion over layers. In Figure (d), we separate documents by a newline token ("\n") and present the distraction proportion of IntraDoc. The results are averaged from 4096 examples. More analysis is presented in Appendix E.

by our experiments in Section 5.1 where we analyse the attention distribution of the models during the language modelling process.

5 Discussion and Analysis

5.1 Can Models Ignore Irrelevant Contexts Before the End-of-Sequence Token?

In the following, we analyse whether models can filter irrelevant context during language modelling by examining the attention score distributions over the context. Specifically, we concatenate two randomly sampled documents from the SlimPajama validation set, separate them by an end-of-sequence token [EOS], and check to which extent the attention distributions of the model focus on the irrelevant document in the sequence. More formally, we define the *distraction proportion* of the token in position p in the current document at layer l as:

$$\text{DistrProp}(l, p) = \sum_{i=1}^{|C_d|} a_{p,i}^{l} \tag{2}$$

where $|C_d|$ denotes the number of tokens in the irrelevant document, $a_{p,i}^{l}$ is the average multi-head attention scores to the i-th token in the irrelevant document C_d at layer l, and $\sum_{i=1}^{|C_d|+p} a_{p,i}^{l} = 1$. In our experiments, we set $|C_d| = 256$, and the results are outlined in Figure 4.

We can see that the latter positions have lower distraction proportions but remain 45%-52% average distraction proportion until the 256th token of the second document, as shown in Figure 4(c). We find that models trained via Bm25Chunk (green line) tend to have lower distraction proportions than other causal masking models, showing that they can better recognise relevant information in the context, matching the results in Figure 3. The above analysis also demonstrates that during the pre-training, causal masking models can be distracted by unrelated documents in context, and the models can be more robust to irrelevant contexts when reducing distractions in pre-training sequences.

Furthermore, in Figure 4(d), we compare IntraDoc and causal masking models using "\n" as the separator instead of [EOS], because [EOS] can only appear at the end of sequences during pre-training using intra-document causal masking. The results indicate that IntraDoc has the lowest distraction proportion compared to causal masking models; meanwhile, Bm25Chunk consistently has a lower distraction proportion than MixChunk and UniChunk using "\n" as the separator. These results match the finding in Section 4.3, where IntraDoc and Bm25Chunk can better recognise relevant information in the context.

5.2 Burstiness Property of Sequences

Chan et al. (2022); Han et al. (2023) found a positive correlation between the model's in-context learning ability and *burstiness* property of the training sequences. Here, burstiness refers to the phenomenon where certain types of tokens occur in clusters or bursts rather than being uniformly distributed across all documents. Burstiness is an inherent property of text; for example, a specific medical term might be frequently used in medical articles and rarely appear in general texts. Higher burstiness results in a lower Zipf's coefficient of token frequency *within a sequence* (Han et al., 2023).

Following Han et al. (2023), we use Zipf's coefficient to measure the burstiness property of pre-training sequences. Formally, let r denote the rank

L	Method	Zipf's Coefficient (α)	In-Context Learning (Acc.)	Knowledge Memorisation (EM)
2K	MixChunk	2.122	63.54	10.33
	UniChunk	2.119	65.02	11.12
	Bm25Chunk	2.107	70.91	11.34
8K	MixChunk	1.976	62.43	7.99
	UniChunk	1.951	65.96	7.92
	Bm25Chunk	1.925	69.47	8.23
2K	IntraDoc	2.119	70.52	11.60
8K	IntraDoc	1.952	71.23	10.99

Table 5: Zipf's coefficients of token frequency in different data. In-context learning and knowledge memorisation abilities are evaluated in Section 4.

of a token in a sequence, and f is a frequency function that maps the rank r to the frequency of that token in the sequence. Then, according to Zipf's law, we have that $f(r;\alpha) \propto \frac{1}{r^{\alpha}}$, where $\alpha \in \mathbb{R}^{+}$ is the Zipf's coefficient; a lower α presents an increased burstiness property within the sequence.

In Table 5, we show the Zipf's coefficients α on different pre-training sequences. Our results show that, for causal masking approaches that use the same chunk size, a lower Zipf's coefficient, which denotes increased burstiness property, often results in more accurate results. However, IntraDoc can obtain significantly better results than UniChunk with the same Zipf's coefficient. The above results indicate that, for causal masking approaches, *the correlation between higher burstiness and better performance could derive from reduced distractions in pre-training chunks*. We report additional evidence for the burstiness property in Appendix D.

Note that duplication in pre-training sequences can also result in increased burstiness property, which may negatively impact the performance of language models. We analyse the distinct n-gram phrases of pre-training sequences in Appendix D and will investigate the impact of duplication using different pre-training corpora in future work.

6 Related Works

Instance-Level Pre-training Data Composition GPT-3 (Brown et al., 2020) was pre-trained by packed documents with causal masking, with the idea that not adopting any dynamic masking can improve pre-training efficiency. Current open-source pre-training frameworks, such as MegatronLM (Shoeybi et al., 2019), FAIRSEQ (Ott et al., 2019), EasyLM (Geng, 2023), LLM360 (Liu et al., 2023b), OLMo (Groeneveld et al., 2024) also follow this strategy for pre-training. In Levine et al. (2022), authors pair similar sentences within the same sequence, while Gu et al. (2023) propose packing documents that contain similar intrinsic tasks for continual pre-training, improving the in-context learning ability of models. Recently, Shi et al. (2023) emphasise that packing relevant documents can enhance language models' in-context learning and context utilisation; however, our findings indicate that packing documents can adversely affect performance, and learning each document independently using intra-document causal masking can reduce the distraction and improve the performance.

Distribution Properties of Pre-Training Data Chan et al. (2022) shows several data distribution properties can drive in-context learning ability, e.g., large numbers of long-tail classes, dynamic meanings of inputs, and Zipf's distribution of class frequency. Han et al. (2023) used a gradient-guided method to select small-scale data for continual pre-training, showing data exhibiting burstiness properties can enhance in-context learning performance.

de Vries (2023) observes that over 80% of pre-training examples consist of fewer than 2K tokens, which suggests that the token distribution of individual documents is not well-suited for the long-context window. To improve the long-context modelling ability of models, Staniszewski et al. (2023) retrieves related documents to construct fine-tuning data with long-range dependency.

Pre-Training Data Quality Gunasekar et al. (2023) selected high-quality data to pre-train a small-size coding model, achieving comparable performance with larger models. Shin et al. (2022); Gao et al. (2021) emphasised the importance of pre-training data diversity. Lee et al. (2022); Tirumala et al. (2023); Soboleva et al. (2023); Abbas et al. (2023) showed the importance of data deduplication on models' generalisation. In our work, we use a diverse and high-quality pre-training dataset, namely SlimPajama (Soboleva et al., 2023), to highlight the importance of the sequence composition strategy on language model pre-training.

7 Conclusion

In this work, we investigate the impact of pre-training sequence composition by pre-training models from scratch. First, we find causal masking can result in unrelated documents distracting language modelling pre-training and hurting the performance on downstream tasks; we show that intra-document causal masking can significantly improve the performance while decreasing the pre-training efficiency. Second, we find improving the relatedness of documents in pre-training chunks for causal masking pre-training can reduce some potential distractions in chunks; our proposed efficient retrieval-based packing method BM25Chunk can significantly improve the performance of language models without reducing pre-training efficiency.

Limitations

Efficiency of Intra-Document Causal Masking We show that intra-document causal masking is an effective method to improve the performance while decreasing the pre-training efficiency. We use FlashAttention2 (Dao, 2023) to implement intra-document causal masking masking without sacrificing too much efficiency (discussed in Appendix A). Still, we do not propose a method to solve this efficiency issue completely.

Objective of Sequences Construction. We discuss sequence construction methods, showing the importance of sequence compositions on the performance of models, but these methods lack an objective during sequence construction. Since specific data distribution properties may be related to models' performance, we will explore using indicators of distributional properties to guide sequence construction in future works.

Scaling The Size of Language Models. Limited by the computation resources, we cannot conduct experiments on larger models with more pre-training steps, and different results might be drawn when increasing the models at a specific scale. However, this work could be directly valuable for investigating pre-training relatively small models that aim at facilitating the use of language models under resource-constrained conditions.

Acknowledgements

PM was partially funded by ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence), EPSRC (grant no. EP/W002876/1), an industry grant from Cisco, and a donation from Accenture LLP; and is grateful to NVIDIA for the GPU donations. YZ was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by UK Research and Innovation (grant EP/S022481/1) and the University of Edinburgh, School of Informatics. This work was supported by the Edinburgh International Data Facility (EIDF) and the Data-Driven Innovation Programme at the University of Edinburgh. The authors extend their gratitude to Xiaomi AI Lab for their GPU donations and assistance; as well as to Tri Dao, Piotr Nawrot, Giwon Hong, Xiaotang Du, and Aryo Gema for their help and feedback.

References

Amro Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *CoRR*, abs/2303.09540.

Francesco Barbieri, José Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1644–1650. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *CoRR*, abs/2304.01373.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Stephanie Chan, Adam Santoro, Andrew K. Lampinen, Jane Wang, Aaditya Singh, Pierre H. Richemond, James L. McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. In *NeurIPS*.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *CoRR*, abs/2307.08691.

Harm de Vries. 2023. In the long (context) run.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn

Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Xinyang Geng. 2023. Easylm: A simple and scalable training framework for large language models.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4849–4870. Association for Computational Linguistics.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *CoRR*, abs/2306.11644.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. 2023. Understanding in-context learning via supportive pretraining data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12660–12673. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Jean Kaddour. 2023. The minipile challenge for data-efficient language models. *CoRR*, abs/2304.08442.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 785–794. Association for Computational Linguistics.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8424–8445. Association for Computational Linguistics.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. 2022. xformers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2022. The inductive bias of in-context learning: Rethinking pretraining example design. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. 2023b. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matteo Pagliardini, Daniele Paliotta, Martin Jaggi, and François Fleuret. 2023. Faster causal attention over large sequences through sparse flash attention. *CoRR*, abs/2306.01160.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Victoria Lin, Noah A Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. 2023. Incontext pretraining: Language modeling beyond document boundaries. *arXiv preprint arXiv:2310.10638*.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woo-Myoung Park, Jung-Woo Ha, and Nako Sung. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5168–5186. Association for Computational Linguistics.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Konrad Staniszewski, Szymon Tworkowski, Yu Zhao, Sebastian Jaszczur, Henryk Michalewski, Lukasz Kuci'nski, and Piotr Milo's. 2023. Structured packing in llm training improves long context utilization. *ArXiv*, abs/2312.17296.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. 2023. D4: improving LLM pretraining via document de-duplication and diversification. *CoRR*, abs/2308.12284.

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

Szymon Tworkowski, Konrad Staniszewski, Mikolaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Milos. 2023. Focused transformer: Contrastive training for context scaling. *CoRR*, abs/2307.03170.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *CoRR*, abs/2305.10429.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

A Implementation of Intra-Document Masking

We use FlashAttention2 (Dao, 2023) to implement intra-document causal masking. The pseudo-code is presented as follows:

Pseudo-code for intra-document causal masking

```
# qkv_states: query, key and value
# max_seqlen: max length of documents
# cu_seqlens: boundaries of documents

qkv_states = qkv_project(hidden_states)

qkv_states = qkv_states.view(batch_size, seq_len, 3,
     num_heads, head_dim)

qkv_states = rotary_embed(qkv_states)

qkv_states = qkv_states.view(batch_size * seq_len, 3,
     num_heads, head_dim)

attn = flash_attn_var_len_qkvpacked_func(qkv_states,
     cu_seqlens, max_seqlen, causal=True)

attn = attn.view(batch_size, seq_len, num_heads *
     head_dim)
attn = output_project(attn)
```

In this implementation of intra-document causal masking, we first apply the rotary position embedding to the hidden states, ensuring INTRADoc uses the same position information that is used in causal masking for each document.

We observe a 4% pre-training speed decrease in our implementation compared to causal masking pre-training, testing on 128 80G A100 GPUs. Another choice to implement intra-document causal masking is using a binary attention bias matrix for masking tokens that belong to other documents. Compared to causal masking using FlashAttention2, we observe that it reduces efficiency by 32% in xFormers (Lefaudeux et al., 2022) when applying the attention bias; besides, it reduces efficiency by 52% using the standard PyTorch implementation.

B Pre-Training Details

Hyperparameters In our experiments, we use the 1.3B model, which has 24 layers, a hidden size of 2048, and 16 attention heads. We use a batch size of 4 million tokens for both the models with 2K and 8K context window sizes and pre-train models using 150B tokens with 38400 steps, which costs 40 hours to pre-training a causal masking model using 128 80G A100 GPUs. We use Adam optimiser with $\beta_1 = 0.90$, $\beta_2 = 0.95$, a weight decay of 0.1, and a cosine learning rate scheduler. The peak learning rate is 3×10^{-4}, decreasing to 3×10^{-5} at the end.

Subset	# documents	Token proportion
CommonCrawl	42960927	52.2%
C4	76520211	26.7%
GitHub	5233374	5.2%
Books	47848	4.2%
ArXiv	383058	4.6%
Wikipedia	7044397	3.8%
StackExchange	7265708	3.3%

Table 6: Pre-training corpus.

Pre-Training Corpus We sample documents with 150B tokens sampled from SlimPajama for pre-training. All models are pre-trained using the same set of documents. In Table 6, we present the number of documents and the token proportions for each subset.

C Analysis of BM25Chunk

C.1 Time Complexity Analysis

In BM25, the similarity score between a query and a document is based on sparse representations, where each query and document is represented by the terms it contains; such sparse representations are stored in *inverted indices*, which map terms to the documents that contain them, along with necessary statistics such as the term frequency and the document frequency. The time complexity of computing similarities between a query and documents in BM25 using an inverted index is $\mathcal{O}(Q \times K)$, where Q denotes the number of tokens in the query, and K represents the number of total documents.

To improve efficiency, we restrict BM25Chunk's retrieval process within a document buffer rather than entire large-scale corpora. The buffer caches k documents, which enables similarity calculations between a term and documents to be at most k times. Since each query is a document, it could contain a large number of tokens; we remove the stop words and randomly sample q tokens to reduce the length. Therefore, the time complexity of sequence construction in BM25Chunk is reduced to $O(q \times k)$. In Figure 5, we test the sequence construction speed using different q and k.

C.2 Implementation Details

We randomly group documents in batches of 5000K and build indexes within each group. The BM25 indexes of pre-training corpora with 150B tokens require 244GB storage memory. For both 2K and 8K settings, the document buffer size k is 3072, and the maximum length of query q is 500. The

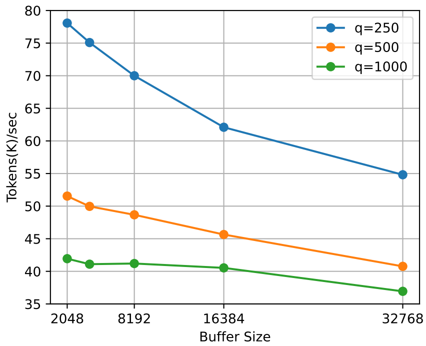


Figure 5: Pre-training sequence construction speeds using different buffer sizes k and maximum query lengths q. Test on 16 CPU cores.

data construction speed is 50.0K tokens per second using 16 CPU cores, and speeds using different settings are presented in Figure 5.

C.3 Ablation Studies

Effectiveness of Document Buffer BM25Chunk conducts retrieval within a document buffer, which may result in retrieving less relevant documents, so we conduct experiments on different document buffer sizes to investigate its effectiveness. We conduct ablation experiments using 0.3B models with a context window of 2048, trained with 13B tokens, the compute-optimal number of tokens according to Hoffmann et al. (2022). We present the PPL improvement over UNIChunk on the validation set of SlimPajama in Table 7. The results show that retrieving from different sizes of document buffers can improve PPL, indicating the effectiveness of retrieving from a small-scale document set. BM25Chunk with a buffer size of 4096 achieves the best result, while increasing the size to 8192 does not improve the PPL.

Effectiveness of Retrieval BM25Chunk conducts multi-hop retrieval to retrieve a sequence of documents, which could potentially help models learn long-distance relationships across documents, and this benefit has been revealed by its high accuracy on HotpotQA, a multi-hop QA task, as shown in Section 4.3. An alternative choice is retrieving multiple documents at once to fill a pre-training chunk, and we present such one-hop retrieval in Table 7. The result indicates that BM25Chunk with multi-hop retrieval can improve the PPL more effectively. Besides, we experiment with random sampling documents from the buffers without retrieval; the result shows the effectiveness of retrieval.

Model (0.3B)	**Document Buffer Size**	**Valid. PPL**
MIXChunk	-	15.474
INTRADoc	-	$12.443_{\downarrow 3.031}$
BM25Chunk	2048	$13.657_{\downarrow 1.817}$
	4096	$\mathbf{12.528}_{\downarrow 2.946}$
	8192	$12.684_{\downarrow 2.790}$
BM25Chunk		
w/o multi-hop retrieval	4096	$13.497_{\downarrow 1.977}$
w/o retrieval	4096	$14.241_{\downarrow 1.233}$
CONTRIEVERChunk	-	$13.720_{\downarrow 1.654}$

Table 7: PPL on the validation set of SlimPajama. Subscript$_{\downarrow}$ is the PPL improvement over MIXChunk. The label "w/o multi-hop retrieval" means retrieving multiple documents at once to construct the sequence; "w/o retrieval" represents random sampling from document buffers, which is equivalent to UNIChunk.

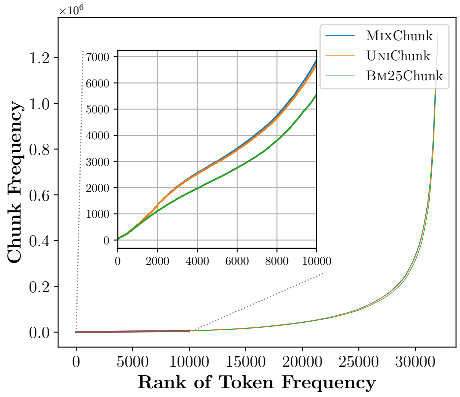


Figure 6: Chunk frequency. The x-axis indicates the frequency rank of tokens; the y-axis presents the number of chunks containing a specific token.

Dense Retrieval Method An alternative retrieval method to BM25 is dense retrieval. We use Contreiver (Izacard et al., 2022) as the dense retriever and compare it with BM25. Following Shi et al. (2023), we embed pre-training documents to dense vectors using Contriever and use FAISS (Johnson et al., 2019) to accelerate the retrieval process instead of using the document buffer. Then, we construct pre-training chunks using the same process introduced in BM25Chunk. We present the result produced by the dense retrieval method in the last line of Table 7. We observe that the improvement of the dense retrieval method is less than BM25.

D Analysis of Data Distribution Properties

Chunk Frequency In addition to Zipf's coefficient, we analyse the burstiness property through the chunk frequency of tokens. Specifically, chunk

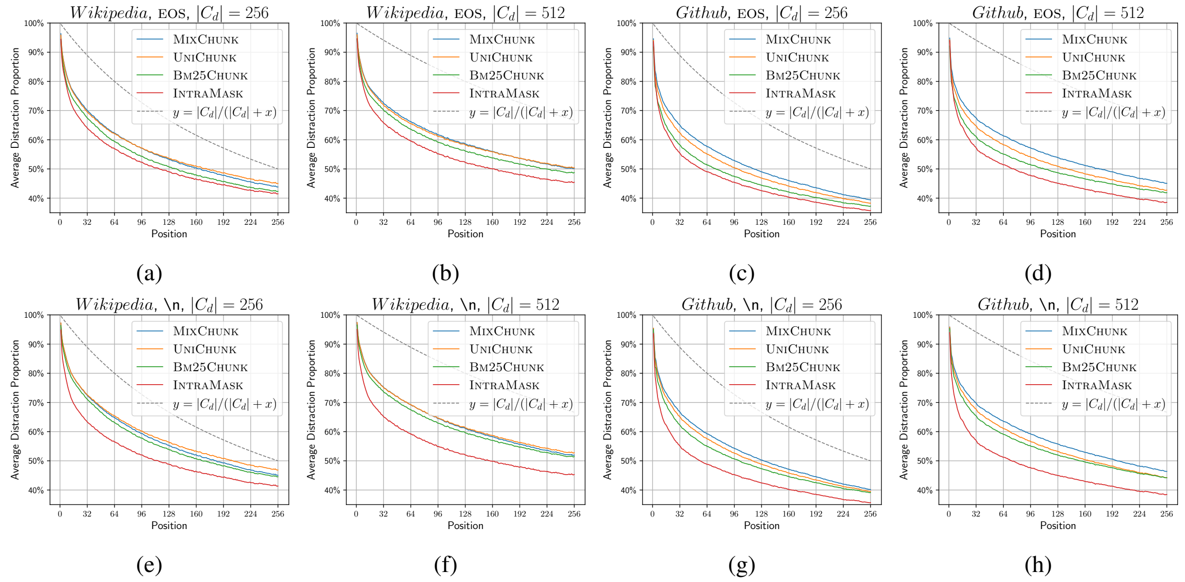


Figure 7: Average distraction proportions over layers. We compare results using different corpora (Wikipedia and GitHub), distraction length ($|C_d| = 256$ and 512), and the separator [EOS] and \n). The first row, (a) (b) (c) and (d), use [EOS] as the separator; the second row, (e) (f) (g) and (h), use \n. The first and the third columns, (a) (c) (e) and (g), have an irrelevant context length $|C_d|$ of 256, and the others are 512. The first two columns, (a) (b) (e) and (f), present the results of Wikipedia, and the last two columns, (c) (d) (g) and (h), present the results of GitHub. We present the baseline $y = |C_d|/(|C_d| + x)$ whose attention scores are uniformly distributed over all preceding tokens.

frequency refers to the number of chunks where a specific token appears. Given a corpus, if a specific token appears in fewer chunks, it indicates more concentrated occurrences in chunks containing the token, demonstrating a higher burstiness property. In Figure 6, we can see that low-frequency tokens appear in fewer chunks in BM25Chunk compared to MIXChunk and UNIChunk, indicating these low-frequency tokens are gathered through the retrieval-based construction method.

Distinct N-gram The burstiness property can correlate to the duplication in a sequence, which may negatively affect models, e.g., models may tend to copy phrases from context. We use SlimPajama, a high-quality and deduplicated dataset, as the pre-training corpus, which can alleviate the duplication issue in BM25Chunk. We use the percentage of distinct n-grams within a sequence to analyse the duplication issue, as shown in Table 10. The results show that, with BM25Chunk, pre-training sequences contain a lower percentage of distinct n-grams than MIXChunk and UNIChunk.

E Analysis of Distraction Proportions in Different Settings

In Figure 7, we report the average distraction proportion (defined in Equation (2)) over layers using different settings. Specifically, we analyse distraction proportions in different settings by varying the *1)* modalities of corpus: text and code using Wikipedia and GitHub; *2)* the separator token: [EOS] and line break token \n; *3)* the length of distraction context, $|C_d| = 256$ and 512.

Method	Δ PPL %	Δ DISTPROP %
MIXChunk	14.6%	3.4%
UNIChunk	15.3%	4.6%
BM25Chunk	13.5%	4.6%
INTRADoc	−0.7%	−0.6%

Table 8: The PPL and DISTPROP changes after replacing the separator [EOS] by \n. A positive value means PPL or DISTPROP increases (performance drops).

Comparing different separators [EOS] and \n, (a) (e), (b) (f), (c) (g), and (d) (h), we observe that causal masking models can obtain lower distraction proportions using [EOS], indicating causal masking models can benefit from [EOS] to ignore irrelevant context during pre-training. We present the impact of changing the separator from [EOS] to \n on PPL and distraction proportion in Table 8. The results show that PPL and DISTPROP increase after the replacement for causal masking models, while INTRADoc obtains better results using \n as the separator since it does not train on sequences

L	Model	CommonCrawl	C4	Wikipedia	GitHub	StackExchange	Book	ArXiv	Avg.
2K	MIXChunk	0.5429	0.4950	0.6238	0.7665	0.5974	0.5001	0.6406	0.5952
	UNIChunk	0.5468	0.4984	0.6298	0.7709	0.6011	0.5033	0.6436	0.5991
	BM25Chunk	<u>0.5496</u>	<u>0.5021</u>	<u>0.6394</u>	<u>0.7782</u>	<u>0.6041</u>	<u>0.5050</u>	<u>0.6452</u>	<u>0.6034</u>
	INTRADoc	**0.5507**	**0.5048**	**0.6426**	**0.7793**	**0.6050**	**0.5062**	**0.6458**	**0.6049**
8K	MIXChunk	0.5402	0.4867	0.6219	0.7443	0.5820	0.5042	0.6531	0.5903
	UNIChunk	0.5429	0.4888	0.6235	0.7483	0.5859	0.5065	0.6564	0.5932
	BM25Chunk	<u>0.5489</u>	<u>0.4952</u>	<u>0.6391</u>	<u>0.7621</u>	<u>0.5919</u>	<u>0.5108</u>	**0.6599**	<u>0.6011</u>
	INTRADoc	**0.5506**	**0.4988**	**0.6443**	**0.7643**	**0.5936**	**0.5119**	<u>0.6597</u>	**0.6033**

Table 9: Evaluation of next token accuracy on SlimPajama's test set.

L	Method	Distinct 2-gram %	Distinct 3-gram %	Distinct 4-gram %
2K	MIXChunk	$71.84_{\pm 14.68}$	$84.06_{\pm 14.47}$	$89.02_{\pm 13.16}$
	UNIChunk	$71.84_{\pm 15.07}$	$84.17_{\pm 14.74}$	$89.16_{\pm 13.26}$
	BM25Chunk	$71.49_{\pm 15.21}$	$84.00_{\pm 14.91}$	$89.07_{\pm 13.41}$
	INTRADoc	$80.35_{\pm 15.26}$	$89.01_{\pm 13.07}$	$92.61_{\pm 11.34}$
8K	MIXChunk	$64.81_{\pm 12.84}$	$80.61_{\pm 13.69}$	$86.76_{\pm 12.76}$
	UNIChunk	$64.57_{\pm 14.09}$	$80.61_{\pm 14.92}$	$86.88_{\pm 13.64}$
	BM25Chunk	$63.49_{\pm 14.63}$	$80.06_{\pm 15.64}$	$86.56_{\pm 14.31}$
	INTRADoc	$79.88_{\pm 14.86}$	$88.90_{\pm 12.63}$	$92.61_{\pm 10.96}$

Table 10: The percentages of the distinct n-grams in different pre-training sequences.

where documents are separated by [EOS] using intra-document causal masking.

Comparing Wikipedia (a) (b) (e) (f) and GitHub (c) (d) (g) (h), MIXChunk is more distracted by the irrelevant context in code generation.

Comparing different length distraction contexts, (a) (b), (c) (d), (e) (f) and (g) (h), models are more distracted when $|C_d|$ increases, while much better than the baseline of uniform distribution $y = |C_d|/(|C_d| + x)$.

Comparing INTRADoc (red line) and causal masking models, we observe that intra-document causal masking results in significantly lower distraction proportions in all cases. This phenomenon may imply that using causal masking without considering the boundaries of documents negatively impacts language modelling performance, and the models can be more robust to irrelevant contexts when increasing the relatedness of documents in pre-training chunks.

F Next Token Accuracy of Pre-Trained Language Models

In addition to PPL, we report the next token accuracy of pre-trained language models in Table 9.