Dwell in the Beginning: How Language Models Embed Long Documents for Dense Retrieval

João Coelho[a,b], **Bruno Martins**[b], **João Magalhães**[c], **Jamie Callan**[a], **Chenyan Xiong**[a]
[a] Language Technologies Institute, Carnegie Mellon University, United States
[b] Instituto Superior Técnico and INESC-ID, University of Lisbon, Portugal
[c] NOVA LINCS, NOVA School of Science and Technology, Portugal
jmcoelho@andrew.cmu.edu

Abstract

This study investigates the existence of positional biases in Transformer-based language models for text representation learning, particularly in the context of web document retrieval. We build on previous research that demonstrated loss of information in the middle of input sequences for causal language models, extending it to the domain of embedding learning. We examine positional biases at multiple stages of the training pipeline for an encoder-decoder neural retrieval model, namely language model pre-training, contrastive pre-training, and contrastive fine-tuning. Experiments with the MS-MARCO document collection reveal that after contrastive pre-training the model already generates embeddings that better capture the beginning of the input content, with fine-tuning further aggravating this effect.

1 Introduction

Recent advancements have allowed Transformer-based models to handle increasingly larger context lengths, resulting in the availability of Language Models (LMs) that can accommodate input lengths reaching tens of thousands of tokens (Xiong et al., 2023). However, studies assessing how well this context is captured by causal LMs (Liu et al., 2023) have shown that models are biased to information contained at the beginning or end of the input, losing information in the middle.

Instead of further analysing text generation, we extend this type of study to text representation learning, which has been a fundamental task for dense retrieval (Xiong et al., 2021; Karpukhin et al., 2020), and is also gaining attention in the context of retrieval-augmented generation (Chevalier et al., 2023; Mu et al., 2023) and recommendation systems (Doddapaneni et al., 2024). Specifically, we focus on web document retrieval, examining how well a single embedding represents a complete web document, while assessing the emergence of eventual position biases.

We start by continuously pre-training and fine-tuning an encoder-decoder model similar to T5-base (Raffel et al., 2020) but with a context length of 2048 tokens, following standard techniques to achieve a model that is representative of the state-of-the-art among the low-parameter scale. We leverage the MS-MARCO (v1) document collection (Nguyen et al., 2016), as this dataset is commonly used in retrieval evaluation benchmarks (Thakur et al., 2021; Muennighoff et al., 2023), and it is one of the major sources of training data for the fine-tuning of neural retrieval models (Zhang et al., 2023; Wang et al., 2022).

We found the existence of a *dwell in the beginning* effect, i.e. a positional bias displayed by the model where earlier parts of the input are dominant in the embedding. We track this behavior by evaluating the model on position-aware tasks during multiple stages of its training. From our experiments, we conclude that these positional biases start emerging during unsupervised contrastive pre-training, and that the heavy reliance on MS-MARCO data for fine-tuning will exacerbate this behavior. Our models and code are available in a public GitHub repository[1].

2 Related Work

Bi-encoders are now the state of the art approach to dense retrieval (Xiong et al., 2021; Karpukhin et al., 2020). Current standard training setups leverage the usage of contrastive loss functions and methods such as ANCE (Xiong et al., 2021) to sample hard negative examples. Other techniques that are often employed include in-domain pre-training (Gao and Callan, 2022) and retrieval-aligned pre-training (Lu et al., 2021; Xiao et al., 2022; Lee et al., 2019; Ma et al., 2022, 2024), which allow for a better fine-tuning starting point, consequently achieving stronger retrieval results.

[1] https://github.com/cxcscmu/LongEmbeddingAnalsys

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 370–377

For long document retrieval, early methods dealt with the increased input length through heuristic aggregation strategies, which rely on segmenting the document into passages that are scored independently, with max-pooling being particularly effective (Dai and Callan, 2019). Instead of aggregating scores, studies like PARADE (Li et al., 2020) considered the aggregation of passage-level representations. Other authors (Boytsov et al., 2022) used Transformer architectures with sparse attention patterns (Beltagy et al., 2020; Zaheer et al., 2020) to model the long inputs more efficiently, showing that, on MS-MARCO, the gains that arise from using such models are limited when compared to simple aggregation strategies.

Currently, LLaRA (Li et al., 2023) achieves state-of-the-art performance in the MS-MARCO document retrieval task, by continually pre-training LLaMA-7B (Touvron et al., 2023) with a retrieval-aligned task. Models like LLaRA leverage context windows of up to 4096 tokens, relying on FlashAttention (Dao et al., 2022; Dao, 2023) for fast and exact full attention computation, together with some variation of Rotary Position Embeddings (RoPE) (Su et al., 2024) or Attention with Linear Biases (ALiBi) (Press et al., 2022). This enables stronger modeling of longer sequences, without the need of additional training, while resorting to full-attention computations.

3 Methodology

This section details the training of a T5-base retriever with 2048 input length (T5-2K), adapting the T5 architecture to follow recent advancements in long-context language modeling, and following a state-of-the-art dense retrieval training pipeline.

3.1 Model Architecture

We use the T5-base architecture as a backbone, replacing the positional embeddings by RoPE (Su et al., 2024). This change was motivated by RoPE's ability to extrapolate to larger contexts, and its compatibility with FlashAttention. Specifically, we use Dynamic NTK-RoPE (Peng et al., 2024), which in theory allows for extrapolation to longer input sequences without further training. The retriever follows a tied bi-encoder architecture, i.e., the same model encodes both queries and documents. The T5 decoder is used as a pooler (Ni et al., 2022), generating a single token and considering its representation as the document embedding.

3.2 Dense Retriever Training Pipeline

Language Modelling Pre-training: Starting from T5-base available at HuggingFace[2], we continuously pre-train the model on 8 billion tokens from the MS-MARCO document collection, for the model to adapt to the new maximum sequence length, new positional embeddings, and MS-MARCO's document distribution. We follow the original T5 span-corruption task, masking 15% of the input sequence, with an average corrupted span length of 3 tokens.

Unsupervised Contrastive Pre-training: In order to align the model with the fine-tuning task, we perform further pre-training following the cropping technique (Izacard et al., 2022). In this task, given a document, a positive pair (s, s^+) is sampled by independently cropping two random spans comprising 10 to 50% of the input. The model is trained to minimize the following contrastive loss:

$$\mathcal{L} = -\frac{1}{n}\sum_i \log \frac{e^{\cos(f(s_i), f(s_i^+))}}{e^{\cos(f(s_i), f(s_i^+))} + \sum_j e^{\cos(f(s_i), f(s_{ij}^-))}}, \quad (1)$$

where each s_i is associated with one positive example s_i^+ as per the sampling technique, and negatives $\{s_{ij}^-\}$ are sampled in-batch. We use a batch size of 128 leveraging GradCache (Gao et al., 2021), and cross-device negatives across 4 GPUs. The representations $f(.)$ generated by the model are compared using the cosine similarity function.

Supervised Contrastive Fine-tuning: We finally fine-tune the model for retrieval in the MS-MARCO dataset for eight epochs. Both the title and body of the documents are used, as this is the default setting for the document retrieval task. We start with ANCE-MaxP negatives (Xiong et al., 2021), refreshing them every two epochs with the model under training. We follow the loss introduced in Equation 1, leveraging labeled query-document pairs. We sample 9 negatives per query, using a batch size of 128 and in-batch negatives. Moreover, cross-device negatives are considered across 4 GPUs, which totals 5120 documents for each query in the batch.

4 Experiments

This section starts by addressing the overall retrieval performance of the T5-2K model. Then, we show the *dwell in the beginning* behavior that is present in the model, investigating each of the training steps to identify its emergence.

[2] https://huggingface.co/t5-base

	Size	MRR@100	R@100
ANCE-MaxP (Xiong et al., 2021)	125M	0.384	0.906
ADORE (Zhan et al., 2021)	110M	0.405	0.919
ICT (Lee et al., 2019)	110M	0.396	0.882
SEED (Lu et al., 2021)	110M	0.396	0.902
RepLLaMA (Ma et al., 2023)	7B	0.456	-
T5-2K (ours)	220M	0.414	0.915

Table 1: Retrieval results on MS-MARCO documents.

4.1 Retrieval Performance

Before moving to the study of the positional biases, we look into the overall performance of our model to assess its soundness, considering the official MS-MARCO evaluation metrics (mean reciprocal rank and recall). For reference, Table 1 contains retrieval results on the MS-MARCO document dataset (development splits), where our model achieves comparable performance to models trained following similar pipelines. The first group references models that do not leverage pre-training tasks, while the ones in the second group incorporate them. Finally, the third group contains a model that also underwent simple fine-tuning, but has 30 times more parameters. Note that other authors have proposed heavily engineered pre-training tasks that do improve results (e.g., COSTA (Ma et al., 2022), Longtriever (Yang et al., 2023), or LLaRA (Li et al., 2023)), but that is out of scope for this work. Appendix A provides additional training details.

4.2 Impact of Relevant Passage Position

For a subset of the queries in the MS-MARCO dataset (i.e., 1130 queries), we can cross-reference their relevant documents with the MS-MARCO passage collection to identify the relevant information within the document through exact matching. In a first experiment assessing the impact of the position of the relevant passage, we retrieve from the collection 11 times: First, a default run with the documents unchanged, followed by 10 runs where the documents associated with the queries have the relevant passage moved to different positions. For each document, given its length l_d and the length of the relevant passage l_p (both in tokens), we compute 10 sequential and uniform insertion points (I_i) for the passage, according to $I_i = (i-1)\frac{l_d - l_p}{9}, i \in \{1, ..., 10\}$, moving the passage from its original position to each I_i.

The performance of our model after one training episode (i.e., before the first ANCE negative refreshing) is depicted in the blue lines of Figure 1. We see that when the relevant passage is moved to the beginning of the document, the performance increases when compared to the default setting (i.e., unchanged documents). Conversely, if the passage is moved anywhere else, the performance drops. The green lines show that the same pattern also holds for RepLLaMA-7B[3] (Ma et al., 2023), i.e. a version of LLaMA-2 fine-tuned for dense retrieval on MS-MARCO for one epoch. In other words, a *dwell in the beginning* effect is observed, where the initial positions are heavily preferred to later ones.

This differs from the *lost in the middle* (Liu et al., 2023) phenomena, where performance would drop significantly only in middle sections, rising in the end. We also note that further fine-tuning on MS-MARCO data will aggravate the behavior, as shown by the orange lines in Figure 1, given the larger performance mismatch between the default setting and insertion positions other than the first.

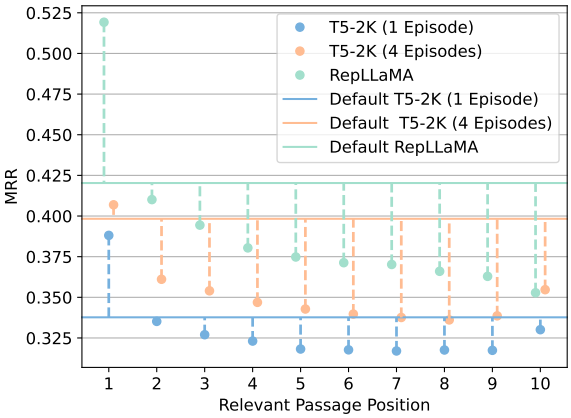

Figure 1: Performance of T5-2K and RepLLaMA. Full lines represent the unchanged version of the documents. Dashed lines represent the variations obtained when the relevant passages are moved to a different position.

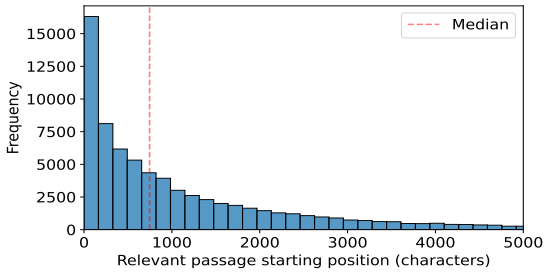

Figure 2: Distribution for the starting position (characters) of relevant passages within 75,000 documents from the MS-MARCO training split.

[3] https://huggingface.co/castorini/repllama-v1-7b-lora-doc

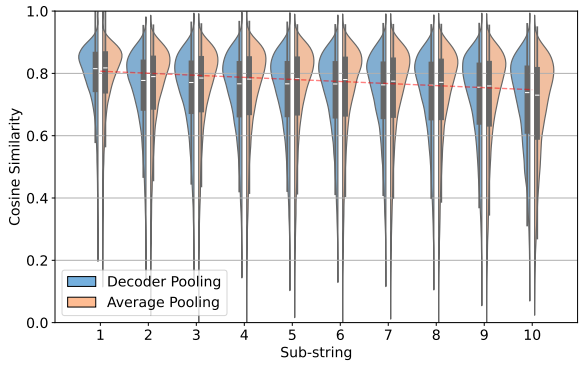

Figure 3: Cosine similarity distribution for exact matching of sub-strings in different locations, using a sample of 24,000 MS-MARCO documents, for the T5-2K model after contrastive pre-training using both decoder-pooling and average-pooling.

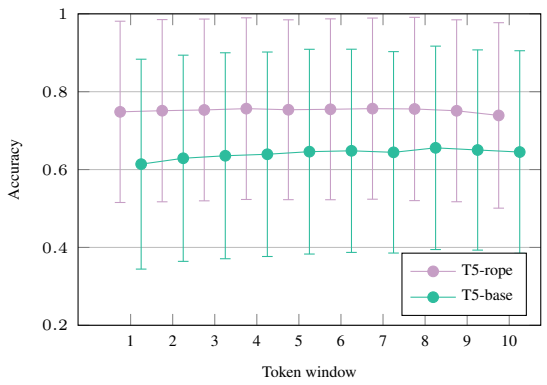

Figure 4: Span prediction accuracy on different zones of the input, using 7000 random 3-token spans per window.

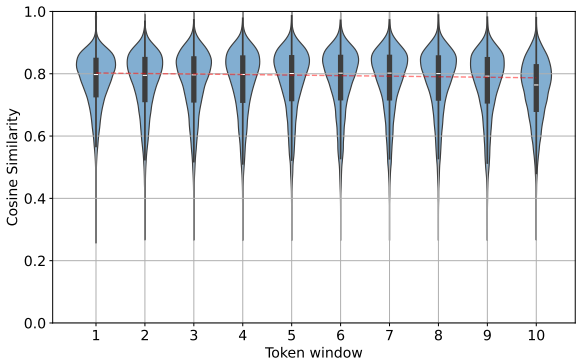

Figure 5: Cosine similarity distribution for exact matching of sub-strings in different locations, using a sample of 24,000 MS-MARCO documents, for the T5-2K model after language model pre-training.

To better understand this behavior, we can look at the distribution in Figure 2, which shows that MS-MARCO documents tend to contain the relevant passage earlier in the document, with the median starting position at 746 characters. This can be impactful for the biases in Figure 1, given the lack of examples with relevant information later in the document. To further investigate this phenomenon, the next sub-sections explore the locality of the pre-training tasks to address potential impacts on long-context modeling.

4.3 Contrastive Pre-training Location Bias

To better estimate positional biases after the contrastive pre-training step, we evaluate the performance of the model on exactly matching sub-strings from different locations. For instance, given a document d, 10 sub-strings are sampled by segmenting d in 10 sequential groups with uniform token length. In other words, the first sub-string contains the first 10% tokens of d, while the last sub-string contains the last 10% tokens. Then, the embedding generated for d is compared with the embedding of each sub-string using the cosine similarity. Figure 3 shows that the similarity values tend to decrease when the position of the sub-string moves from the beginning, and that this behavior holds for strategies that either use decoder pooling or average pooling of token representations.

This indicates that the representation generated for a document is better at capturing its earlier contents. While in the previous sub-section similar behavior could be justified by the data's underlying distribution, the pseudo-queries and documents for this task were independently sampled from the same uniform distribution over the input. This suggests that the bias is intrinsic to models trained on web documents, e.g. by fitting to information distributions commonly found in real web documents that follow the *inverted pyramid* writing style (Koupaee and Wang, 2018), where earlier paragraphs are often more representative. Since web documents are the most common source of contrastive pre-training data (Wang et al., 2022; Izacard et al., 2022), this is problematic for tasks where the whole input must be accurately captured, as is for instance the case of retrieval augmented generation (Chevalier et al., 2023; Mu et al., 2023).

4.4 Span Corruption Location Bias

Finally, we look into the language model pre-training task. We evaluate on the original task, by independently corrupting spans of 3 tokens across multiple parts of the input, divided in ten windows as per the previous experiments. Through this, we can see if the accuracy of the model varies when

predicting the correct spans across the different parts of the input document.

Figure 4 shows uniform performance, suggesting no inherent bias in this task using RoPE. We also evaluate the original T5-base, and see that although it shows a slightly higher performance on predicting later positions, it is still rather uniform. As none of the models display the *dwell in the beginning* effect, we conclude that the language modeling pre-training task did not induce any biases, and that this behavior emerged as soon as the embedding task was added to the training pipeline. To further solidify this result, Figure 5 shows the evaluation of the T5-2K model after language model pre-training (but before embedding-based learning) on the embedding task from Section 4.3, showing a similar pattern to Figure 4, without a noticeable *dwell in the beginning* effect.

5 Conclusions and Future Work

This study investigated a *dwell in the beginning* effect on Transformer-based models for document retrieval. Through experiments with a T5 model and RepLLaMA, we observed that the embeddings tend to favor information located at the beginning of the input, leading to decreased performance when relevant information is elsewhere in the document. We investigate each step in the training pipeline, namely language model pre-training, contrastive pre-training, and contrastive fine-tuning, showing that biases emerge in the contrastive pre-training step, and that they persist throughout the fine-tuning process. Our findings emphasize the importance of considering the quality of embeddings for long inputs, particularly in contexts where effectively capturing the entire sequence is essential for the downstream task. Moreover, our results can further justify previous research which showed limited gains on long-sequence modeling for MS-MARCO, when compared to aggregation approaches (Boytsov et al., 2022).

As for future work, we note that while our experiments focused on tied encoders, a similar study can be conducted using untied weights, given the size mismatch between queries and documents. Furthermore, addressing the identified biases may involve devising more robust pre-training tasks, or curating better-distributed datasets, all while considering evaluation on appropriate retrieval benchmarks that require long-context modeling (Wang et al., 2023; Saad-Falcon et al., 2024).

Limitations and Ethical Considerations

All the datasets and models used in our experiments are publicly available, and we provide the source code that allows for reproduction of the results, as well as model checkpoints.

By using large pre-trained language models, we acknowledge the risks associated with the presence of inherent biases embedded within the models, which may inadvertently perpetuate or amplify societal biases present in the training data.

One limitation in the work reported on this paper relates to the fact that our tests have only used English data. Other languages can expose different phenomena in terms of how document-context is handled, and future work can perhaps consider other datasets such as the one from the NeuCLIR competition (Lawrie et al., 2024). Doing a similar analysis on other domains besides web documents would also be interesting, and we encourage the research community to further study document-context modeling in connection to different types of information retrieval tasks.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. This research was supported by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (i.e., the Center For Responsible AI), and also by the Fundação para a Ciência e Tecnologia (FCT), specifically through the project with reference UIDB/50021/2020 (DOI: 10.54499/UIDB/50021/2020), the project with reference UIDP/04516/2020 (DOI: 10.54499/UIDB/04516/2020), and also through the Ph.D. scholarship with reference PRT/BD/153683/2021 under the Carnegie Mellon Portugal Program.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *ArXiv*, abs/2004.05150.

Leonid Boytsov, Tianyi Lin, Fangwei Gao, Yutian Zhao, Jeffrey Huang, and Eric Nyberg. 2022. Understanding Performance of Long-Document Ranking Models through Comprehensive Evaluation and Leaderboarding. *ArXiv*, abs/2207.01262.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting Language Models to

Compress Contexts. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *International Conference on Research and Development in Information Retrieval (SIGIR 2019)*.

Tri Dao. 2023. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. *ArXiv*, abs/2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Annual Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Sumanth Doddapaneni, Krishna Sayana, Ambarish Jash, Sukhdeep S. Sodhi, and Dima Kuzmin. 2024. User Embedding Model for Personalized Language Prompting. *ArXiv*, abs/2401.04858.

Luyu Gao and Jamie Callan. 2022. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. In *Annual Meeting of the Association for Computational Linguistics (ACL 2022)*.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup. In *Workshop on Representation Learning for NLP*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.

Mahnaz Koupaee and William Yang Wang. 2018. WikiHow: A Large Scale Text Summarization Dataset. *ArXiv*, abs/1810.09305.

Dawn J. Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldaini, and Eugene Yang. 2024. Overview of the TREC 2023 NeuCLIR Track. *ArXiv*, abs/2404.08071.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Annual Meeting of the Association for Computational Linguistics (ACL 2019)*.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: passage representation aggregation for document reranking. *ArXiv*, abs/2008.09093.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making Large Language Models A Better Foundation For Dense Retrieval. *ArXiv*, abs/2312.15503.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. *ArXiv*, abs/2307.03172.

Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is More: Pretrain a Strong Siamese Encoder for Dense Text Retrieval Using a Weak Decode. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*.

Guangyuan Ma, Xing Wu, Zijia Lin, and Songlin Hu. 2024. Drop your Decoder: Pre-training with Bag-of-Word Prediction for Dense Passage Retrieval. *ArXiv*, abs/2401.11248.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction. In *International Conference on Research and Development in Information Retrieval (SIGIR 2022)*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. *ArXiv*, abs/2310.08319.

Jesse Mu, Xiang Lisa Li, and Noah D. Goodman. 2023. Learning to Compress Prompts with Gist Tokens. *ArXiv*, abs/2304.08467.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: massive text embedding benchmark. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *Findings of the Association for Computational Linguistics (ACL 2022)*.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. YaRN: Efficient Context Window Extension of Large Language Models. In *International Conference on Learning Representations (ICLR 2024)*.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations (ICLR 2022)*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*.

Jon Saad-Falcon, Daniel Y. Fu, Simran Arora, Neel Guha, and Christopher Ré. 2024. Benchmarking and Building Long-Context Retrieval Models with LoCo and M2-BERT. *ArXiv*, abs/2402.07440.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Annual Conference on Neural Information Processing Systems (NeurIPS 2021)*.

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. *ArXiv*, abs/2302.13971.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2023. DAPR: A benchmark on document-aware passage retrieval. *ArXiv*, abs/2305.13915.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *ArXiv*, abs/2212.03533.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations (ICLR 2021)*.

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. *ArXiv*, abs/2309.16039.

Junhan Yang, Zheng Liu, Chaozhuo Li, Guangzhong Sun, and Xing Xie. 2023. Longtriever: a Pre-trained Long Text Encoder for Dense Document Retrieval. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *International Conference on Research and Development in Information Retrieval (SIGIR 2021)*.

Xin Zhang, Zehan Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, and Min Zhang. 2023. Language Models are Universal Embedders. *ArXiv*, abs/2310.08232.

A Training Details

This appendix starts by detailing the training setup used in our experiments, and it then presents experimental results that further assess the impact of the different training stages.

A.1 Hyperparameters

The following subsections detail the hyperparameters used for model training. If a certain element is not stated, the default value from the HuggingFace Trainer API was used. All models were trained in the same computational infrastructure with 4 NVIDIA A100 40GB GPUs.

A.1.1 Span Corruption Pre-training

Optimizer	AdamW
Initial learning rate	1e-5
Scheduler	Cosine
Batch size	80
Gradient accumulation	16
Gradient clipping	1
Weight decay	0
Total steps	49152
Warm-up steps	10%

Table 2: Set of hyperparameters considered for span-corruption pre-training.

A.1.2 Contrastive Pre-training

Optimizer	AdamW
Initial learning rate	5e-6
Scheduler	Linear
Batch size	128
Gradient accumulation	1
Gradient cache chunk size	24
Hard negatives per query	0
Epochs	1

Table 3: Set of hyperparameters considered for contrastive pre-training.

A.1.3 Fine-tuning

Optimizer	AdamW
Initial learning rate	5e-6
Scheduler	Linear
Batch size	128
Gradient accumulation	1
Gradient cache chunk size	24
Hard negatives per query	9
Epochs	8

Table 4: Set of hyperparameters considered for final model fine-tuning.

A.2 Impact of Each Training Step

Table 5 aligns our training pipeline with previous work, showing the importance of the pre-training tasks, and the benefits of multiple fine-tuning steps with negative refreshing. Note that the performance without any pre-training is particularly low since the model had no previous exposure to the new rotary embeddings.

LM Pre-training	Contrastive Pre-training	Fine-tuning	MRR	R@100
✗	✗	1 episode	0.177	0.632
✓	✗	1 episode	0.350	0.872
✓	✓	1 episode	0.372	0.889
✓	✓	4 episodes	0.414	0.915

Table 5: Performance on MS-MARCO for different combinations of pre-training tasks, and after fine-tuning.