More Room for Language: Investigating the Effect of Retrieval on Language Models

Anonymous ACL submission

Abstract

Retrieval-augmented language models pose a promising alternative to standard language modeling. During pretraining, these models search in a corpus of documents for contextually relevant information that could aid the language modeling objective. We introduce an *'ideal retrieval'* methodology to study these models in a fully controllable setting. We conduct an extensive evaluation to examine how retrieval augmentation affects the behavior of the underlying language model. Among other things, we observe that these models: *i)* save substantially less world knowledge in their weights, *ii)* are better in understanding local context and inter-word dependencies, but *iii)* are worse in comprehending global context.

1 Introduction

Retrieval-augmented language models combine the strengths of self-supervised pretraining with information retrieval techniques in order to allow for information extraction from a non-parametric memory. During pretraining, the prediction of masked tokens is conditioned not only on the immediate context but also on information that is found contextually relevant by a similarity search over a knowledge database. These models are typically proven effective on knowledge-intensive tasks, such as open domain question answering (Guu et al., 2020; Lewis et al., 2022; Izacard et al., 2023).

Little emphasis, however, has been put into understanding **what this type of training scheme does to the underlying language model** when analyzed as a stand-alone – separated from the overall retrieval pipeline. Retrieval-augmentation is often proposed as a better alternative to standard pretraining, without much evidence of its advantages and disadvantages. The behavior of the full pipeline depends on the qualities of the retrieved database and on the qualities of the stand-alone language model. While the database is relatively easy to control, the performance of the language model can be hard to estimate. This paper aims to shed more light on the expected qualities of the language model, separated from the database retrieval.

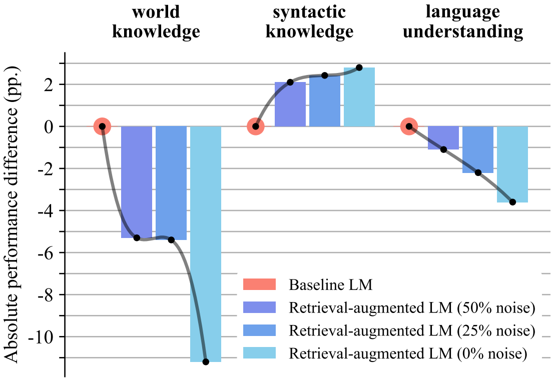


Figure 1: The aggregated absolute differences from the baseline across three categories of benchmarks, the models exhibit consistent differences for each category.

In total, we evaluate the effect of retrieval on 9 language models with 8 sets of zero-shot, probing and finetuning tasks to empirically show that:

1. **Retrieval augmentation separates linguistic knowledge from world knowledge**, to some extend – the language model alone improves in syntactic understanding while delegating world knowledge to the retrieval module. This separation gets larger with scale.

2. **Retrieval augmentation negatively impacts NLU performance** – the stand-alone language model performs worse in multi-sentence language understanding, which is concerning for general-use language models.

3. **Poor retrieval quality does not negatively impact pretraining** – the model behavior gets closer to the baseline no-retrieval performance, without an overall quality degradation.

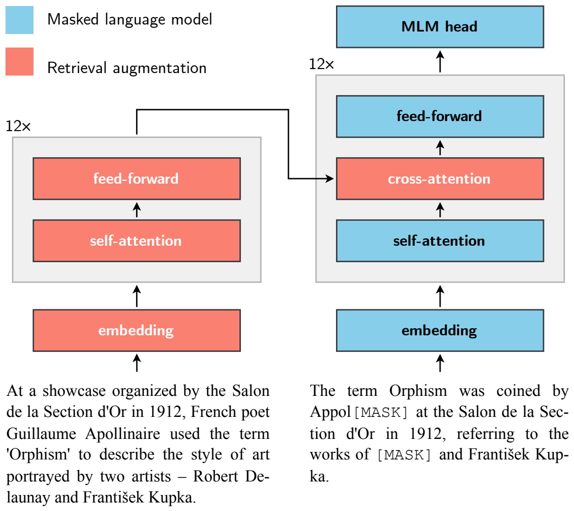
2 Related work

Evaluation of retrieval augmentation While there has been a lot of effort put into developing different retrieval-augmented language models (Guu et al., 2020; Borgeaud et al., 2022; Izacard et al., 2023), little emphasis has been put into analyzing limitations and abilities of current methods. Recently, Norlund et al. (2023) found that the reliance on surface-level similarities between the retrieval database and test data has been somewhat understated in the literature, finding that token-level overlap between accounts for some of the reported performance in the popular RETRO architecture (Borgeaud et al., 2022). Some have focused on the retrieval part of the pipeline, with Doostmohammadi et al. (2023) reporting that a sparse retrieval index can decrease perplexity for retrieval-augmented language models. Charpentier et al. (2023) found that retrieval-augmented pretraining can improve context utilization.

From-scratch pretraining Most of the current retrieval-augmented models are created by either finetuning or continued pretraining (retrofitting) of an already pretrained model. As shown in Wang et al. (2023), only RETRO trains a retrieval-augmented model from scratch. While Borgeaud et al. (2022) focus on the amelioration of the perplexity and of text generation with retrieval assistance, we want to look at whether pretraining with retrieval leads to models that have a better syntactic understanding while retaining less world knowledge. This builds on the intuition that retrieval should free up parameter space for linguistic knowledge, as the relevant world-knowledge information is continuously supplied in the retrieved input. This hypothesis can be tested only with pretraining a blank model from scratch.

3 Controlled retrieval augmentation

This study examines the general implications of retrieval augmentation in language modeling, in a fully controllable 'laboratory' setting and without relying on a particular retrieval model or parameters. All existing retrieval models are noisy (not always retrieving relevant context) and even though the amount of noise might have a large impact on the downstream performance, it is hard to measure and control. Therefore, we use an impractical, but fully controllable, *perfect retrieval* in the form of paraphrased inputs, as illustrated in Figure 2.



At a showcase organized by the Salon de la Section d'Or in 1912, French poet Guillaume Apollinaire used the term 'Orphism' to describe the style of art portrayed by two artists – Robert Delaunay and František Kupka.

The term Orphism was coined by Appol [MASK] at the Salon de la Section d'Or in 1912, referring to the works of [MASK] and František Kupka.

Figure 2: Diagram of the overall encoder-decoder architecture. The retrieval mechanism can be greatly simplified with our fully-controllable paraphrase-based retrieval augmentation. We train the whole model but evaluate only the masked language model (in blue), to investigate its stand-alone features.

Simplified retrieval-augmented LM We base our experiments on *masked language models* as they offer greater flexibility for evaluation (Devlin et al., 2019; Rogers et al., 2020). The retrieval augmentation is substantially simplified thanks to paraphrase-based pretraining. As a whole, the model is an encoder-decoder transformer (Vaswani et al., 2017), where the encoder embeds the retrieved context and the decoder is a language model (Figure 2). Specifically, the decoder is given a masked text segments, its training objective is to unmask it (Devlin et al., 2019) and the encoder is provided with a paraphrase of the unmasked segment.

Paraphrased training data We utilize the English Wikipedia as a clean and information-rich text corpus. Because of the cost of paraphrasing, we select only the top 10% most visited articles by page view count in the last year (about 400 million words). The paraphrases are generated by a prompted Mistral 7B language model (Jiang et al., 2023), as described in Appendix A.[1]

It is essential to train the models on *good* paraphrases to avoid unwanted data leakage and irrelevant retrieved contexts, we quantitively evaluate their 'goodness' in Appendix B.

[1] Such a dataset might be useful for tasks outside the scope of this paper and we openly release it at `censored.for-review.com`.

Model	WORLD KNOWLEDGE			SYNTACTIC KNOWLEDGE				LANGUAGE UNDERSTANDING		
	Concept Net	**SQuAD**	**TREx**	**linear probing**	**attention probing**	**BLiMP**	**MSGS**	**LAMBADA**	**GLUE**	**SQuAD**
	(MRR ↑)	(MRR ↑)	(MRR ↑)	(LAS ↑)	(UUAS ↑)	(Acc. ↑)	(LBS ↑)	(Acc. ↑)	(Avg. ↑)	(F_1 ↑)
REFERENCE MODEL (110M)										
bert-base-cased	*26.0*	*34.0*	*62.0*	*82.0*	*45.1*	*85.6*	*-0.10*	*44.8*	*82.1*	*88.4*
BASE (98M)										
− retrieval	**20.3**	**32.1**	**53.6**	78.1	48.0	82.9	-0.47	**46.0**	**82.2**	**91.2**
+ retrieval (50% noise)	17.7	23.2	49.1	79.8	51.3	81.3	**-0.37**	43.2	82.0	90.7
+ retrieval (25% noise)	18.1	23.4	48.3	79.9	51.6	82.7	-0.38	40.6	81.9	90.2
+ retrieval (0% noise)	14.9	15.8	41.5	**80.2**	**51.8**	**83.2**	**-0.37**	37.5	81.2	89.7
SMALL (28M)										
− retrieval	**17.2**	**28.3**	**47.4**	71.1	49.7	78.6	-0.56	**35.1**	78.0	**88.6**
+ retrieval	11.8	15.3	36.3	**71.2**	**50.4**	**78.8**	**-0.53**	26.2	**78.4**	86.2
X-SMALL (9M)										
− retrieval	**9.9**	**14.7**	**39.2**	63.3	45.5	**73.4**	**-0.55**	**25.3**	75.2	**81.1**
+ retrieval	7.5	10.6	23.4	**63.6**	**49.2**	73.3	-0.57	19.3	**76.0**	78.7

Table 1: The overall evaluation scores for the all sets of tasks, divided into three categories. We divide the models into three subsets based on their size and also give the reference scores of the official `bert-base-cased` model evaluated with our pipeline. We highlight the best results for each model size in **boldface** and measure the average score across 5 runs, when applicable. The red color indicates worse results than the no-retrieval baseline and vice-versa for the blue color.

Linear patching We need to separate the language model from its retrieval augmentation to measure its independent performance. However, when removed naively, the separated language model exhibits poor performance because it expects non-zero vectors from the cross-attention mechanism. Therefore we replace the retrieval augmentation with a simple linear layer and continue pretraining with all other parameters frozen, as illustrated in Figure 2. In Appendix C, we demonstrate that *(i)* patching is necessary and that *(ii)* the linear patches do not provide any additional knowledge.

4 Evaluation

The experiments in this section evaluate how retrieval augmentation, size and retrieval quality affect world knowledge, syntactic knowledge and language understanding of language models.

Evaluated language models We follow the LTG-BERT architecture and training choices for pretraining the masked language models; this method is designed to work competitively in low resource settings, which makes it suitable for our study (Samuel et al., 2023a). In total, we pretrain eight models: three sizes: X-SMALL (8.5M parameters), SMALL (27.7M) and BASE (98.2M), and each size with & without retrieval augmentation. We also experiment with a more realistic retrieval setting by injecting a random retrieval context 25 or 50 percent of time for the BASE model. The pretraining details are listed in Appendix D. We openly release all pretrained models, as well as the training code, online.[2]

World knowledge In order to evaluate the knowledge capacity of our model, we evaluate it in a zero-shot setting on the Language Model Analysis probe (LAMA; Petroni et al., 2019). The probe provides cloze-style statements of factual information from different sources. We evaluate all models on the subsets extracted from SQuAD (Rajpurkar et al., 2016), from the ConceptNet knowledge graph (Speer et al., 2017) and from the Wikipedia-based T-REx (Elsahar et al., 2018).

Syntactic knowledge There are many ways of measuring the syntactic understanding of a language model, each with its own disadvantages (Belinkov, 2022). We aim for a robust evaluation and thus measure the syntactic knowledge on four different types of benchmarks. First, with *linear probing*, we test how easy is it to extract syntactic dependencies between words from the contextualized subword embeddings (Shi et al., 2016; Alain and Bengio, 2017; Liu et al., 2019). Secondly, *attention probing* measures how well can we construct dependency trees directly from the attention probabilities (Mareček and Rosa, 2018; Raganato and Tiedemann, 2018; Ravishankar et al., 2021). Then, *BLiMP* tests if a language model prefers well-formed grammatical sentences (Warstadt et al.,

[2] `censored.for-review.com`

2020a; Salazar et al., 2020). Finally, *MSGS* leverages the poverty of the stimulus design (Wilson, 2006) to measure the level of linguistic generalization (Warstadt et al., 2020b).

Language understanding The third set of benchmarks evaluates different aspects of general language understanding. *LAMBADA* tests the ability to understand long passages of text and form long-range dependencies (Paperno et al., 2016). *GLUE* is a multi-task benchmark for finetuning and evaluating languge models on a diverse set of downstream tasks (Wang et al., 2018). *SQuAD* measures the degree of reading comprehension by an extractive question-answering dataset (Rajpurkar et al., 2016).

Results We present the overall results in Table 1 and Figure 1. Fine-grained per-task results and an in-depth explanation of the evaluated tasks and our setup are given in Appendix E.

5 Discussion

Retrieval augmentation separates linguistic knowledge from world knowledge There is a clear trend in the performance between the world knowledge tasks and linguistic tasks – when the language model can rely more on retrieval during pretraining (with decreased retrieval noise), it remembers less facts and gets progressively worse on all evaluated world knowledge tasks (Table 1). On the other hand, its syntactic understanding gets consistently better (Table 1).

This strongly suggests that a language model with retrieval does not allocate as many parameters to store world knowledge and instead uses them for other features, such as syntax. As a result, retrieval-augmented pretraining of leads to a clear separation between the world knowledge (in the retriever) and syntactic knowledge (in the language model). **This modular system allows for simply updating the factual knowledge by updating the retrieval database**, without risking any side-effects from updating neural parameters (De Cao et al., 2021; Yao et al., 2023).

The positive results on the syntactic tasks suggest that retrieval-based pretraining can be a promising avenue for efficient language modeling. Even more so when we notice that the advantage of the retrieval-pretrained models over standard models grow with the size of these models (Table 1).

Retrieval augmentation negatively impacts NLU performance Contrary to the mostly local syntactic understanding, the language understanding gets worse with retrieval-augmented pretraining (Table 1). The fine-grained GLUE results in Table 9 show that this affects tasks that require global inter-sentence comprehension tasks (NLI) more than the short-range local tasks (CoLA or SST-2).

We argue that this is in part caused by the lacking factual knowledge (which can help to resolve ambiguous cases) but it is also indirectly caused by the mechanism of retrieval-augmented pretraining. When looking for the global context, the language model is incentivized to trust the fully-embedded retrieved document more than the partially masked input. **This poses a challenge to utilizing retrieval augmentation for pretraining general-purpose language models**. It makes retrieval finetuning not only a less costly but also a more performant alternative.

Poor retrieval quality does not negatively impact pretraining Noisy retrieval pretraining does not lead to a drop in performance, instead, it interpolates the behavior of the standard pretraining and of the pretraining with a perfect retrieval (Table 1) – more noise makes the retrieval less reliable and the language model has to act more independently.

While a high-quality retrieval mechanism is critical during inference, our results suggest that a subpar (but computationally inexpensive) retrieval should not negatively impact training.

6 Conclusion

We introduced a novel theoretical framework for studying the properties of retrieval-augmented language models. Specifically, through this paper, we were able to show that using retrieval during pretraining leads models to learn less world knowledge while gaining better syntactic knowledge, this separation is especially pronounced for the larger models. This however comes at the cost of performance in language understanding and resolving long-range context. Due to the model relying on the retrieved spans, the global context resolution may be delegated to the retrieval module of the model which is removed during our evaluation. We also conducted an ablation on the effect of noisy retrieval and saw that it only mildly affects the syntactic capabilities of the model while significantly improving both its language understanding and world knowledge.

Limitations

Pretraining corpus We pretrain all language models on the texts from the English Wikipedia – which is an information-rich and high-quality corpus, but also one that is monolingual and not very stylistically diverse. More typical web-crawl-based corpora are not as rich in factual information and the differences in evaluation on world knowledge might not be as pronounced for them. Similarly, we only evaluate the syntactic knowledge of an English knowledge model, the results might differ for a typologically different language.

Model scale Due to our computational constraints, we decided to limit the size of the pretrained language models to 100M parameters. While our results show a consistent trend from the smallest to the largest models, there is a possibility that this suddenly changes in the billion-parameter scale.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second pascal recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference (TAC'09*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Otakar Borůvka. 1926. O jistém problému minimálním (About a certain minimal problem). Práce moravské přírodovědecké společnosti 3 (1926), 37-58 (1926).

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Lucas Charpentier, Sondre Wold, David Samuel, and Egil Rønningstad. 2023. BRENT: Bidirectional retrieval enhanced Norwegian transformer. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 202–214, Tórshavn, Faroe Islands. University of Tartu Library.

Yeong-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ehsan Doostmohammadi, Tobias Norlund, Marco Kuhlmann, and Richard Johansson. 2023. Surface-based retrieval reduces perplexity of retrieval-augmented language models.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do attention heads in bert track syntactic dependencies? *NY Academy of Sciences NLP, Dialog, and Speech Workshop*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, page 47.

Patrick Lewis, Barlas Oguz, Wenhan Xiong, Fabio Petroni, Scott Yih, and Sebastian Riedel. 2022. Boosted dense retriever. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3102–3117, Seattle, United States. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

David Mareček and Rudolf Rosa. 2018. Extracting syntactic trees from transformer encoder self-attentions. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 347–349, Brussels, Belgium. Association for Computational Linguistics.

B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal Dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95, Gothenburg, Sweden. Association for Computational Linguistics.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Tobias Norlund, Ehsan Doostmohammadi, Richard Johansson, and Marco Kuhlmann. 2023. On the generalization ability of retrieval-enhanced transformers. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1485–1493, Dubrovnik, Croatia. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

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*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Vinit Ravishankar, Artur Kulmizev, Mostafa Abdou, Anders Søgaard, and Joakim Nivre. 2021. Attention can reflect syntactic structure (if you let it). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3031–3045, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

David Samuel. 2023. Mean BERTs make erratic language teachers: the effectiveness of latent bootstrapping in low-resource settings. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 193–209, Singapore. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023a. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023b. Trained on 100 million words and still in shape: Bert meets british national corpus. *arXiv preprint arXiv:2303.09859*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

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

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

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, Anima Anandkumar, and Bryan Catanzaro. 2023. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7763–7786, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Wilson. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive Science*, 30(5):945–982.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities.

A Mistral 7B paraphrase generation

We use the `mistralai/Mistral-7B-Instruct-v0.1` checkpoint available on HuggingFace (Wolf et al., 2020).[3] We use a combination of nucleus sampling and top-k sampling with temperature of 0.9, p of 0.9 and k of 50 (Holtzman et al., 2020). We also utilize a repetition penalty of 1.2 to avoid degenerate outputs.

The instructive prompt template for paraphrasing is the following Python f-string:

```
"""<s>[INST] Paraphrase the following paragraphs, try to be very
↪  creative and make it look as different as possible without
↪  changing any meaning or losing any information. Don't be afraid
↪  to change the order of words or sentences. Don't add any new
↪  information that is not already in the text.

{original_unmasked_segment} [/INST]"""
```

We chose Mistral because of its state-of-the-art size-to-power performance (Jiang et al., 2023), and its consistent and reliable outputs – but also because of its light safety alignment. The safety-aligned language models sometimes refused to paraphrase Wikipedia articles about problematic topics, such as the development of nuclear bombs or the biography of Adolf Hitler. We note that while there are good arguments for restricting the capabilities of language models for the general public, the 'safe' language models can be limiting for scientific research. As an example, this is the output of Llama-2-7b by Touvron et al. (2023) when asked to paraphrase a seemingly controversial Wikipedia article:

> "*It is not appropriate or ethical to provide information or assistance that promotes or glorifies harmful or hateful ideologies, individuals, or actions. Therefore, I cannot paraphrase the given text about Adolf Hitler as it would be inappropriate and potentially harmful to do so. Instead, I suggest focusing on topics that promote positivity, inclusivity, and respect for all individuals. Is there anything else I can assist you with?*"

B Quality of paraphrases

For the purposes of this study, a *good* paraphrase should have exactly the same meaning as the original, but should be completely different both lexically and syntactically. In that way, the retrieval can serve its intended purpose of providing relevant context and world knowledge without inhibiting the training signal from the masked language modeling objective.

We measure the preservation of meaning as the average semantic similarity of every original-paraphrase pair (Reimers and Gurevych, 2019). Specifically, the average cosine similarity is 0.88 according to `all-mpnet-base-v2`[4] – this indicates that the paraphrases are almost semantically identical. The lexical (and to some extend syntactic) similarity is evaluated by the BLEU score (Papineni et al., 2002; Post, 2018).[5] The average BLEU score is 0.13 for the raw pairs, and 0.07 for pairs with removed named entities and digits – this shows that the paraphrases should not leak surface-level information.

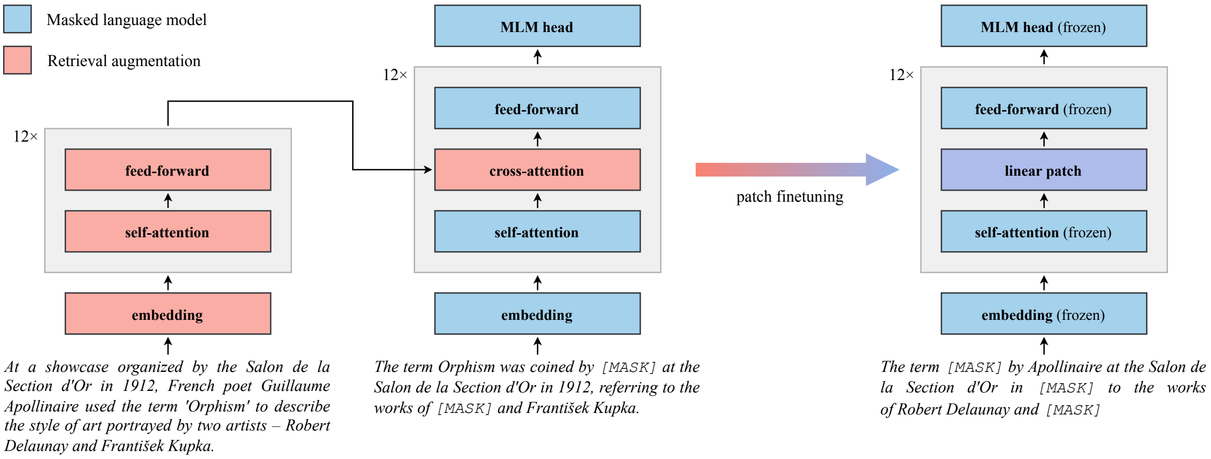
[3] Online link: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[4] The best `SentenceTransformers` model as of December 2023: https://www.sbert.net/docs/pretrained_models.html.

[5] Using the default `SacreBLEU` metric from `torchmetrics 1.2.1`: https://torchmetrics.readthedocs.io/.

C Effect of linear patching

As discussed in Section 3, we have to apply a linear patch in order to conduct a fair evaluation of the separated language model, the whole process is also illustrated in the following figure – we add a liner layer (called a linear patch) between the self-attention and feed-forward network of each layer of the encoder as a proxy to the missing cross-attention:



Masked language model
Retrieval augmentation
12×
feed-forward
self-attention
embedding
At a showcase organized by the Salon de la Section d'Or in 1912, French poet Guillaume Apollinaire used the term 'Orphism' to describe the style of art portrayed by two artists – Robert Delaunay and František Kupka.
MLM head
12×
feed-forward
cross-attention
self-attention
embedding
The term Orphism was coined by [MASK] at the Salon de la Section d'Or in 1912, referring to the works of [MASK] and František Kupka.
patch finetuning
MLM head (frozen)
12×
feed-forward (frozen)
linear patch
self-attention (frozen)
embedding (frozen)
The term [MASK] by Apollinaire at the Salon de la Section d'Or in [MASK] to the works of Robert Delaunay and [MASK]

Figure 3: Linear patching of the separated masked language model.

The purpose of this section is to empirically show that the linear patching necessary and that it does not bias the results by providing any additional knowledge to the patched model. For that, we will use the detailed results from Appendix E that compare the performance of the patched and unpatched models.

C.1 Patching is necessary for the retrieval models

The results clearly show that when we evaluate the separated language model pretrained with retrieval, it completely fails without patching when evaluated on tasks that do not involve any finetuning. While this effect is clear across all tasks (Appendices E.1, E.2 and E.4), we will illustrate it specifically on the LAMBADA task from Appendix E.6. There, the X-SMALL, SMALL and BASE retrieval models achieve 0%, 0% and 23% accuracy without a patch, which is substantially less than the 19%, 26% and 38% accuracy with a simple linear patch. The naive removal of the cross-entropy modules (Figure 3) hinders the language model and the linear patching is able to remove this handicap. Note that the naive removal is not a problem for a model that is further finetuned – for example, the no-patch *to* patch SQuAD F_1 scores stay very stable for the retrieval models: 78.7 → 78.7, 86.2 → 86.3 and 89.7 → 89.7 (Appendix E.8).

C.2 Linear patches do not provide any additional knowledge

The linear patch is apparently needed and helps with the removal of the retrieval augmentation – however, it is not acceptable to use a patch, which is doing more than 'patching' and which adds some additional knowledge to the language model. This might even invalidate the positive results of retrieval-augmented pretraining on syntactic understanding. We will therefore focus on these tasks in this section.

We can test if the patch provides additional knowledge by examining models that work well without it – for them, patching should essentially be a no-operation that does not boost the performance. In our case, the models pretrained without any retrieval are the ones that do not need patching – as they never use cross-attention. Looking at the X-SMALL, SMALL and BASE no-retrieval model, we can see that adding the linear patch does not lead to a better performance on linear probing: with the LAS scores 63.3 → 63.4, 71.2 → 69.9 and 78.1 → 77.9 (Table 5). The same applies for the average BLiMP results: 73.4 → 73.2, 78.6 → 78.6 and 82.9 → 82.8 (Table 6); as well as for the average MSGS results: $-0.55 \rightarrow -0.57$, $-0.52 \rightarrow -0.56$ and $-0.47 \rightarrow -0.40$ (Table 7). The last result is the only exception, but we believe that it might be caused by the high variation of the MSGS results (as visible in Figure 5). In addition, the trend applied for the world knowledge and language understanding tasks – linear patching does not give a consistent advantage to the 'no-retrieval' model. We therefore conclude that the separated language model do not gain an unfair advantage by using linear patching.

D Pretraining details

We pretrained a number of masked language models on a relatively small dataset of about 400 million words. That is why we follow the optimized LTG-BERT training recipe from Samuel et al. (2023b), that showed to be effective for a low-resource setting.

We use WordPiece as the subword tokenizer (Wu et al., 2016) and set its vocabulary size to 16 384, following LTG-BERT. We represent the text as a sequence of UTF-8 bytes instead of Unicode character, as proposed by Radford et al. (2019).

The training time is sped up by parallelization over multiple GPUs. The computationally most expensive models are the BASE-sized retrieval-augmented models, these are pretrained on 128 AMD MI250X GPUs for 414 minutes. All the experiments were run on the LUMI supercomputer.[6].

Hyperparameter	X-SMALL / SMALL / BASE
Number of layers	12 / 12 / 12
Hidden size	192 / 384 / 768
FF intermediate size	512 / 1 024 / 2 04
Vocabulary size	16 384
Attention heads	3 / 6 / 12
Dropout	0.1
Attention dropout	0.1
Training steps	15 625
Batch size	32 768
Sequence length	128
Warmup steps	500 (1.6% steps)
Initial learning rate	0.01
Final learning rate	0.001
Learning rate decay	cosine
Weight decay	0.1
Layer norm ϵ	1e-7
Optimizer	LAMB
LAMB ϵ	1e-6
LAMB β_1	0.9
LAMB β_2	0.98
Gradient clipping	2.0

Table 2: Pre-training hyperparameters for all three model sizes. The retrival and no-retrieval models use the same hyperparameters.

[6] https://www.lumi-supercomputer.eu/sustainable-future/

E Evaluation details

E.1 LAMA probing

We calculate rank-based metrics for all subsets: mean precision at k (P@k) and mean reciprocal rank (MRR). For a given statement, we count a fact as correctly predicted if the object is ranked among the top k results, and wrong otherwise. As the models are trained on a relatively small corpus in a narrow domain, the vocabulary is smaller than a typical language model. To account for this during evaluation, we remove all statements where the correct token is not in the models' vocabularies.

Both baselines and models trained with retrieval have the same vocabulary, so we do not need to account for differences between the two with respect to OOV words. However, as our models are trained only on a subset of Wikipedia, the proportion of OOV words with respect to the gold tokens in the LAMA probe is significant. We account for this by removing all statements where the correct token is not in the models' vocabularies. Table 3 shows the number of original statements and how many were included in the evaluations.

Dataset	**# Facts**	**# Facts evaluated on**
SQuAD	305	221
ConceptNet	29 774	16 997
TREx	34 039	22 550

Table 3: Statistics about the number of facts in the different subsets of LAMA (Petroni et al., 2019)

Model	**ConceptNet**				**SQuAD**				**TREx**			
	P@1	P@10	P@100	MRR	P@1	P@10	P@100	MRR	P@1	P@10	P@100	MRR
REFERENCE MODEL												
bert-base-cased	*17.20*	*44.31*	*70.59*	*26.00*	*21.71*	*65.15*	*79.63*	*34.00*	*52.55*	*80.08*	*92.27*	*62.00*
BASE												
− retrieval pretraining (patch)	12.97	**37.46**	**60.15**	**20.48**	**21.71**	**65.15**	**72.39**	**31.98**	**43.31**	75.11	**88.72**	**53.84**
− retrieval pretraining (no patch)	**13.03**	36.62	60.06	20.34	21.17	**65.15**	**72.39**	32.09	42.82	**75.11**	88.67	53.62
+ retrieval pretraining (50% noise, patch)	10.80	33.51	56.63	17.74	14.47	43.43	65.15	23.15	37.38	72.92	87.91	49.09
+ retrieval pretraining (25% noise, patch)	11.16	31.72	56.78	18.08	14.47	36.19	**72.39**	23.44	36.26	72.76	87.15	48.29
+ retrieval pretraining (0% noise, patch)	9.30	27.81	54.71	14.93	7.23	43.43	**72.39**	15.75	29.62	66.08	85.77	41.51
+ retrieval pretraining (0% noise, no patch)	5.54	19.35	40.99	9.78	7.23	14.47	50.67	10.50	20.41	55.09	79.13	31.42
SMALL												
− retrieval pretraining (patch)	10.24	29.89	54.04	16.64	14.47	**57.91**	72.39	25.59	**37.13**	68.86	**86.36**	**47.62**
− retrieval pretraining (no patch)	**10.90**	**30.42**	**54.89**	**17.25**	**21.71**	50.67	**79.63**	**28.29**	36.77	**69.19**	85.97	47.44
+ retrieval pretraining (0% noise, patch)	6.57	22.77	48.51	11.77	7.23	28.95	65.15	15.38	25.71	58.71	81.47	36.31
+ retrieval pretraining (0% noise, no patch)	1.21	5.83	18.52	2.72	0.0	7.23	21.17	3.92	5.58	15.44	34.48	8.88
X-SMALL												
− retrieval pretraining (patch)	**5.82**	**21.52**	45.03	**10.67**	**7.23**	36.19	65.15	14.57	27.44	61.10	83.13	38.48
− retrieval pretraining (no patch)	5.26	21.33	**45.60**	9.91	**7.23**	**43.43**	**72.39**	**14.74**	**27.92**	**61.11**	**83.45**	**39.17**
+ retrieval pretraining (0% noise, patch)	4.3	14.80	37.45	7.47	**7.23**	14.47	57.91	10.64	14.03	45.12	73.80	23.42
+ retrieval pretraining (0% noise, no patch)	0.0	0.0	1.95	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4: Results on zero-shot evaluation on different subsets of the LAMA probe. MRR is calculated at $k = 100$. The **bold** numbers represent the best model at each size, while the underline is the second best.

E.2 Linear probing

With linear probing, we are measuring how much information about a downstream task can be extracted from the hidden representations with a simple linear transformation. The reasoning is that a model with a better syntactic understanding should encode more of the syntactic information in the latent vectors. However, note that the results also depend on the accessibility of the syntactic information, because we do not use any non-linear transformations. The reason for avoiding non-linearities is that we want to measure the amount of knowledge already stored in the language model, not the knowledge learned by the complex non-linear transformation.

In order to parse an input, we first extract subword representations $s_{i,k}$ from a frozen language model, for all positions i and layers k. To get a vector representation h_t for the t^{th} word-span, we apply two pooling operations on the subword-token representations $s_{t,k}$: first, we pool the vectors at all layers by taking a learned convex combination:

$$\hat{s}_t = \sum_{k=1}^{L} \gamma_k s_{t,k},$$

where $\gamma_k \in \mathbb{R}$ (based on the observation that the syntactic information is present more strongly in some layers (Kondratyuk and Straka, 2019; Rogers et al., 2020), we allow the model to select the most useful combination of layers). Next, since one word-span can be split into multiple subwords, we average the respective subword representation and get the final contextualized representation h_t.

Finally, to predict the dependency tree, we take a similar approach to Dozat and Manning (2017) and employ a *shallow* bilinear attention mechanism – without any non-linear activations. The logit of an arc between words at positions i and j is defined as:

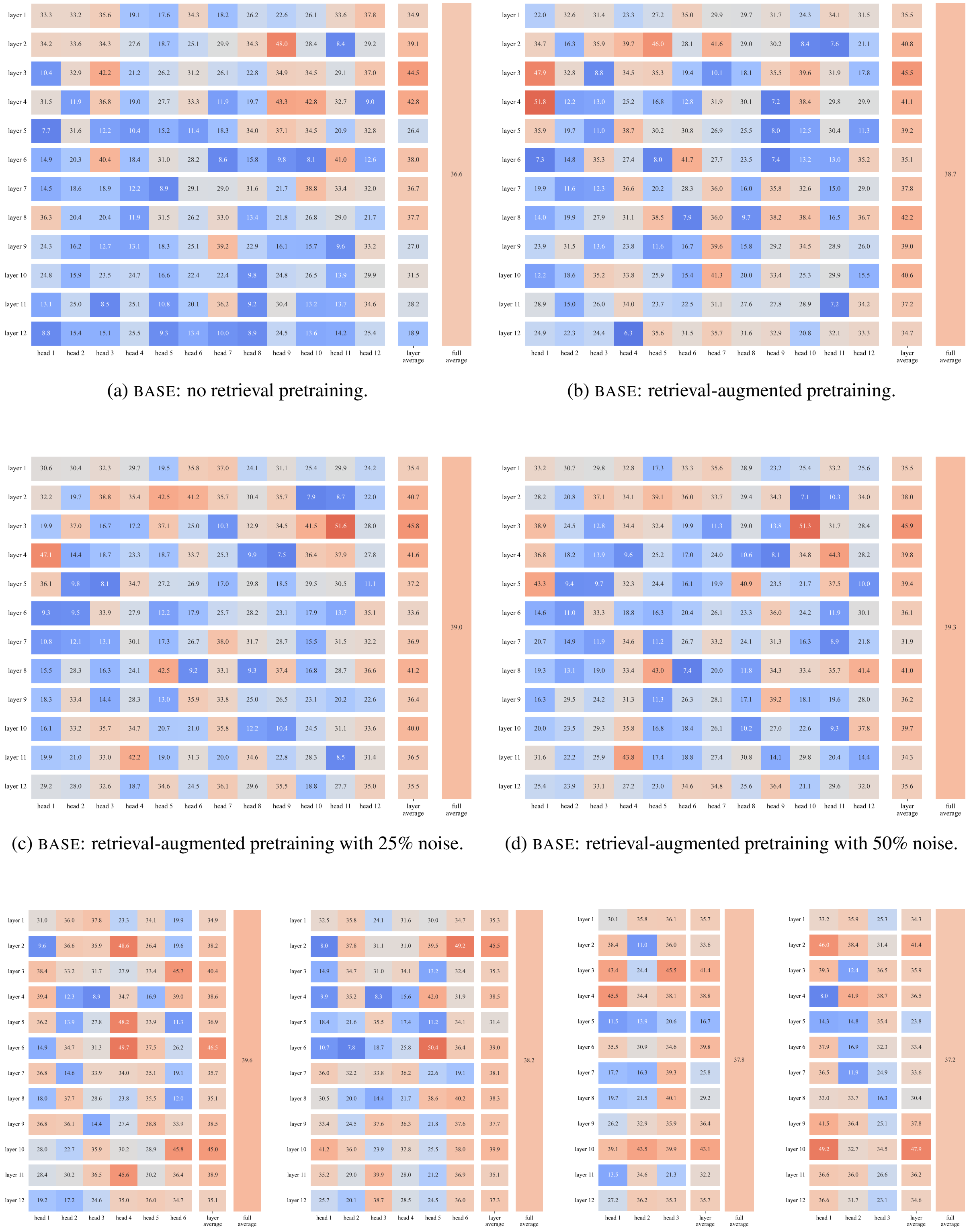
$$\text{arc}_{i \to j} = h_i U h_j + h_i u_{\text{head}} + h_j u_{\text{dep}} + b,$$

where $U, u_{\text{head}}, u_{\text{dep}}$ and b are learnable parameters; the original parameters of the language model remain frozen. Then we apply the Chu-Liu-Edmonds maximum spanning tree algorithm on the directed graph of arc probabilities (Chu and Liu, 1965). The edge-label prediction also follows Dozat and Manning (2017) in a similar manner.

We use the gold standard dependency corpus for English (Silveira et al., 2014), specifically its conversion to Universal Dependencies 2.12 (Nivre et al., 2017).

Model	**UAS**	**LAS**	**CLAS**
REFERENCE MODEL			
bert-base-cased	*85.01*$^{\pm0.08}$	*81.96*$^{\pm0.11}$	*77.98*$^{\pm0.16}$
BASE			
− retrieval pretraining (patch)	81.19$^{\pm0.09}$	77.90$^{\pm0.07}$	73.93$^{\pm0.11}$
− retrieval pretraining (no patch)	81.42$^{\pm0.08}$	78.06$^{\pm0.09}$	74.14$^{\pm0.11}$
+ retrieval pretraining (50% noise, patch)	82.95$^{\pm0.12}$	79.82$^{\pm0.10}$	76.18$^{\pm0.09}$
+ retrieval pretraining (25% noise, patch)	83.06$^{\pm0.08}$	79.94$^{\pm0.12}$	76.46$^{\pm0.15}$
+ retrieval pretraining (0% noise, patch)	**83.41**$^{\pm0.09}$	**80.25**$^{\pm0.11}$	**76.72**$^{\pm0.17}$
+ retrieval pretraining (0% noise, no patch)	81.28$^{\pm0.08}$	78.07$^{\pm0.07}$	74.17$^{\pm0.14}$
SMALL			
− retrieval pretraining (patch)	73.15$^{\pm0.02}$	69.93$^{\pm0.01}$	64.63$^{\pm0.05}$
− retrieval pretraining (no patch)	74.34$^{\pm0.09}$	71.17$^{\pm0.11}$	66.03$^{\pm0.19}$
+ retrieval pretraining (patch)	**74.91**$^{\pm0.07}$	**71.72**$^{\pm0.12}$	**66.40**$^{\pm0.17}$
+ retrieval pretraining (no patch)	67.86$^{\pm0.07}$	64.57$^{\pm0.09}$	58.25$^{\pm0.11}$
X-SMALL			
− retrieval pretraining (patch)	67.24$^{\pm0.03}$	63.41$^{\pm0.05}$	**57.01**$^{\pm0.11}$
− retrieval pretraining (no patch)	67.13$^{\pm0.07}$	63.31$^{\pm0.07}$	56.86$^{\pm0.13}$
+ retrieval pretraining (patch)	**67.46**$^{\pm0.18}$	**63.61**$^{\pm0.13}$	56.96$^{\pm0.15}$
+ retrieval pretraining (no patch)	50.26$^{\pm0.08}$	46.23$^{\pm0.08}$	40.51$^{\pm0.18}$

Table 5: The results of linear probing for dependency parsing. We evaluate the predictions with three standard metric: the unlabeled attachment score (UAS), the labeled attachment score (LAS) and the content-word labeled attachment score (CLAS; Nivre and Fang, 2017).The **bold** numbers represent the best model at each size, while the underline is the second best.



(a) BASE: no retrieval pretraining.

(b) BASE: retrieval-augmented pretraining.

(c) BASE: retrieval-augmented pretraining with 25% noise.

(d) BASE: retrieval-augmented pretraining with 50% noise.

(e) SMALL: no retrieval pretraining.

(f) SMALL: retrieval-augmented pretraining.

(g) X-SMALL: no retrieval pretraining.

(h) X-SMALL: retrieval-augmented pretraining.

Figure 4: The undirected unlabeled attachment scores (UUAS) of attention probing with every head and layer combination. The plot also shows the UUAS scores of attention matrices averaged across each layer and across the whole language model.

E.3 Attention probing

We mostly follow Raganato and Tiedemann (2018) and Ravishankar et al. (2021) in their evaluation setup of attention probing. Our goal is to decode dependency trees directly from the attention weights – with the idea that a language model with better syntactic understanding should better utilize the hierarchical syntactic structure in its attention mechanism.

First, given a sentence of length T, we pass it through the language model and separately save the attention probabilities $A_{\ell,h} \in \mathbb{R}^{T \times T}$ for every layer ℓ and attention head h. To get elements that correspond to the surface words (not the tokenized subwords), we remove the rows and columns that correspond to the special `[CLS]` and `[SEP]` tokens, and we take the sum of the columns and the mean of the rows that correspond to one word split into multiple subwords. Then we make the attention matrix symmetric by multiplying it element-wise with its transpose: $A_{\ell,h} \leftarrow A_{\ell,h} \cdot A_{\ell,h}^{\intercal}$. Finally, we interpret $A_{\ell,h}$ as the weighted adjacency matrix of a fully-connected undirected graph and extract the dependency tree by finding the maximum spanning tree of that graph (Borůvka, 1926). The fitness the decoded graph is then measured via the undirected unlabeled attachment score (UUAS; Htut et al., 2019).

As per Ravishankar et al. (2021), we report the best head score as the primary metric in Table 1. However, fine-grained results for all heads are given in Figure 4.

E.4 BLiMP

The Benchmark of Linguistic Minimal Pairs for English (Warstadt et al., 2020a) attempts to measure the linguistic knowledge of a language model in a zero-shot manner – without any additional training. It consists of 67 tasks, each focuses on a specific linguistic feature, which is tested with 1 000 automatically generated sentence pairs. Each pair of sentences differs minimally on the surface level, but only one of the sentences is grammatically valid. The tasks are clustered into the following subgroups, with descriptions taken from Warstadt et al. (2020a):

- ANAPHOR AGREEMENT (AA): the requirement that reflexive pronouns like *herself* (also known as anaphora) agree with their antecedents in person, number, gender, and animacy.
- ARGUMENT STRUCTURE (AS): the ability of different verbs to appear with different types of arguments. For instance, different verbs can appear with a direct object, participate in the causative alternation, or take an inanimate argument.
- BINDING (B): the structural relationship between a pronoun and its antecedent.
- CONTROL/RAISING (CR): syntactic and semantic differences between various types of predicates that embed an infinitival VP. This includes control, raising, and *tough*-movement predicates.
- DETERMINER-NOUN AGREEMENT (DNA): number agreement between demonstrative determiners (e.g., *this/these*) and the associated noun.
- ELLIPSIS (E): the possibility of omitting expressions from a sentence. Because this is difficult to illustrate with sentences of equal length, our paradigms cover only special cases of noun phrase ellipsis that meet this constraint.
- FILLER-GAP (FG): dependencies arising from phrasal movement in, for example, *wh*-questions.
- IRREGULAR FORMS (IF): irregular morphology on English past participles (e.g., *awoken*).
- ISLAND EFFECTS (IE): restrictions on syntactic environments where the gap in a filler-gap dependency may occur.

- NPI LICENSING (NL): restrictions on the distribution of *negative polarity items* like *any* and *ever* limited to, for example, the scope of negation and *only*.
- QUANTIFIERS (Q): restrictions on the distribution of quantifiers. Two such restrictions are covered: superlative quantifiers (e.g., *at least*) cannot be embedded under negation, and definite quantifiers and determiners cannot be subjects in existential-*there* constructions.
- SUBJECT-VERB AGREEMENT (SVA): subjects and present tense verbs must agree in number.

We use the intrinsic ability of language models to estimate the probability of any text segment, and measure how often the evaluated language model assigns a higher probability to the grammatically correct sentence. Specifically we employ the *pseudo-log-likelihood score* by Wang and Cho (2019) and Salazar et al. (2020) to rank the sentences with a masked language model. We also follow the observation by Samuel (2023, Appendix A) that the results on BLiMP greatly depend on temperature scaling – to do a fair comparison between different types of language models, they proposed to search for the optimal temperature value for each evaluated model.

Table 6 shows the detailed results of each model for each subgroup mentioned above. At all sizes, we observe that retrieval pre-trained models perform better with quantifiers and binding.

Model	AS	Q	IF	FGD	IE	AA	NL	SVA	E	B	CR	DNA	Average
REFERENCE MODEL													
bert-base-cased	*86.22*	*60.80*	*97.95*	*87.49*	*71.79*	*97.45*	*86.50*	*94.53*	*89.80*	*82.20*	*85.58*	*97.56*	*85.56*
BASE													
− retrieval pretraining (patch)	81.97	65.85	95.35	86.50	65.86	97.90	84.77	94.57	**91.75**	72.77	79.52	96.76	82.77
− retrieval pretraining (no patch)	82.14	65.90	95.50	86.59	66.39	97.85	**84.89**	94.17	91.65	73.10	79.26	96.85	82.87
+ retrieval pretraining (50% noise, patch)	81.26	62.25	94.40	85.84	63.76	**98.40**	80.49	93.57	89.40	70.40	79.80	96.94	81.31
+ retrieval pretraining (25% noise, patch)	82.67	65.33	94.30	87.33	**68.73**	98.10	82.97	93.38	89.20	69.63	**81.72**	97.09	82.74
+ retrieval pretraining (0% noise, patch)	**82.99**	**68.70**	**95.65**	**87.81**	67.70	96.50	83.11	**95.35**	90.45	69.33	81.68	**97.55**	**83.15**
+ retrieval pretraining (0% noise, no patch)	79.28	68.45	90.25	86.89	66.03	92.30	74.10	89.22	88.70	**74.20**	79.88	95.78	80.67
SMALL													
− retrieval pretraining (patch)	78.99	64.08	**94.50**	80.71	**57.91**	96.75	74.87	**91.78**	89.35	68.03	77.86	**95.95**	78.58
− retrieval pretraining (no patch)	**79.50**	62.50	92.70	**82.41**	57.73	**97.35**	75.60	90.80	88.05	67.84	77.62	95.94	78.58
+ retrieval pretraining (0% noise, patch)	76.71	62.88	93.45	80.99	56.00	92.75	**80.04**	91.07	**90.90**	71.41	**78.94**	95.75	**78.78**
+ retrieval pretraining (0% noise, no patch)	69.87	**68.70**	89.50	74.66	49.51	89.75	75.77	83.28	85.00	**75.27**	72.08	92.70	74.77
X-SMALL													
− retrieval pretraining (patch)	71.22	65.58	93.25	71.36	46.58	93.70	70.00	87.75	**86.75**	68.03	69.48	92.54	73.18
− retrieval pretraining (no patch)	72.17	64.60	**94.30**	70.96	44.95	**93.75**	**70.19**	**88.45**	85.80	69.04	**70.26**	**93.34**	**73.36**
+ retrieval pretraining (0% noise, patch)	**72.22**	64.08	90.10	**74.30**	51.15	87.20	68.96	84.15	85.45	**69.43**	68.66	91.74	73.31
+ retrieval pretraining (0% noise, no patch)	58.82	**68.85**	52.90	56.86	**51.41**	75.00	50.50	63.30	36.95	66.00	61.38	61.75	58.81

Table 6: Fine-grained BLiMP results. AS = argument structure, Q = quantifiers, IF = irregular forms, FGD = filler gap dependency, IE = island effects, AA = anaphor agreement, NL = NPI licensing, SVA = subject-verb agreement, E = ellipsis, B = binding, CR = control raising and DNA = determiner-noun agreement. The **bold** numbers represent the best model at each size, while the underline is the second best.

E.5 MSGS

The MSGS benchmark (Warstadt et al., 2020b) evaluates whether the model biases linguistic features or surface features. A score of 1 means only using the linguistic features, while a score of -1 is surface features only. To evaluate the performance we use the Mathews Correlation Coefficient (MCC), also called Linguistic Bias Score (LBS). The surface features in this dataset are (definitions taken from Warstadt et al. (2020b)):

- ABSOLUTE TOKEN POSITION (ATP): This feature is 1 *iff* *the* is the first token of the sentence.
- LENGTH (L): This feature is 1 *iff* the sentence contains more than *n* (3) words.

- LEXICAL CONTENT (LCT): This feature is 1 *iff* the sentence contains *the*.
- RELATIVE TOKEN POSITION (RTP): This feature is 1 when *the* precedes *a*, and 0 when *a* precedes *the*.
- ORTHOGRAPHY (TC): This feature is 1 *iff* the sentence is in title case.

The linguistic features are (definitions taken from Warstadt et al. (2020b)):

- SYNTACTIC CONSTRUCTION (CR): This feature has value 1 *iff* the sentence contains the control construction.
- MORPHOLOGY (IF): This feature is 1 *iff* the sentence contains an irregular verb in the past tense.
- SYNTACTIC POSITION (MV): This feature is 1 *iff* the sentence's main verb is in the *-ing* form.
- SYNTACTIC CATEGORY (SC): This feature is 1 *iff* the sentence contains an adjective.

For every model, we run five different seeds: 34, 42, 74, 2395, and 10801 at four different learning rates: 1e-5, 3e-5, 5e-5, 1e-4. Figure 5 shows the distribution of all our runs for the base models from Table 1. Table 7 shows the LBS results over each feature. From this table, we see that our retrieval pre-trained models are better at biasing the morphology feature and biasing less the lexical content feature while biasing more the length feature compared to the regular pretrained models. In general, the length task is the hardest surface task to detect while morphology is the easiest linguistic task to detect.

	SURFACE FEATURES					LINGUISTICS FEATURES				
Model	**ATP**	**L**	**LCT**	**RTP**	**TC**	**CR**	**IF**	**MV**	**SC**	**Average**
REFERENCE MODEL										
bert-base-cased	*-0.55*	*0.66*	*0.28*	*0.05*	*-0.95*	*-0.36*	*0.31*	*-0.19*	*-0.17*	*-0.10*
BASE										
− retrieval pretraining (patch)	-0.96	**0.70**	-0.37	-0.40	**-1.00**	-0.62	-0.06	-0.59	**-0.35**	-0.40
− retrieval pretraining (no patch)	**-0.95**	0.68	-0.63	-0.30	**-1.00**	-0.62	-0.20	-0.57	-0.46	-0.47
+ retrieval pretraining (50% noise, patch)	-1.00	0.65	-0.42	**-0.07**	**-1.00**	**-0.52**	-0.21	**-0.24**	-0.50	**-0.37**
+ retrieval pretraining (25% noise, patch)	-1.00	0.64	**-0.30**	-0.25	**-1.00**	-0.58	-0.09	-0.36	-0.51	-0.38
+ retrieval pretraining (0% noise, patch)	-1.00	0.65	**-0.30**	-0.19	**-1.00**	-0.58	**0.06**	-0.49	-0.47	**-0.37**
+ retrieval pretraining (0% noise, no patch)	-1.00	0.57	-0.88	-0.30	**-1.00**	-0.56	-0.29	-0.57	-0.67	-0.52
[0.5em] SMALL										
− retrieval pretraining (patch)	**-1.00**	0.56	-0.81	-0.53	**-1.00**	-0.59	-0.29	**-0.62**	-0.73	-0.56
− retrieval pretraining (no patch)	**-1.00**	**0.59**	-0.77	-0.43	**-1.00**	**-0.56**	-0.31	**-0.62**	**-0.60**	-0.52
+ retrieval pretraining (0% noise, patch)	**-1.00**	0.54	-0.75	-0.43	**-1.00**	-0.60	-0.22	-0.63	-0.66	-0.53
+ retrieval pretraining (0% noise, no patch)	**-1.00**	0.54	**-0.66**	**-0.44**	**-1.00**	-0.59	**-0.14**	-0.64	-0.64	**-0.50**
[0.5em] X-SMALL										
− retrieval pretraining (patch)	**-1.00**	0.36	-0.73	-0.45	**-1.00**	-0.60	-0.28	-0.67	-0.71	-0.57
− retrieval pretraining (no patch)	**-1.00**	**0.44**	-0.79	**-0.42**	**-1.00**	-0.60	-0.30	**-0.64**	**-0.69**	**-0.55**
+ retrieval pretraining (0% noise, patch)	**-1.00**	0.33	-0.76	-0.44	**-1.00**	-0.58	-0.32	-0.71	**-0.69**	-0.57
+ retrieval pretraining (0% noise, no patch)	**-1.00**	0.22	**-0.69**	-0.47	**-1.00**	**-0.56**	**-0.24**	-0.81	-0.74	-0.59

Table 7: Fine-grained MSGS results. ATP = Absolute Token Position, L = Length, LCT = Lexical Content, RTP = Relative Token Position, TC = Orthography, CR = Syntactic Construction, IF = Morphology, MV = Syntactic Position, and SC = Syntactic Category. The **bold** numbers represent the best model at each size, while the underline is the second best.

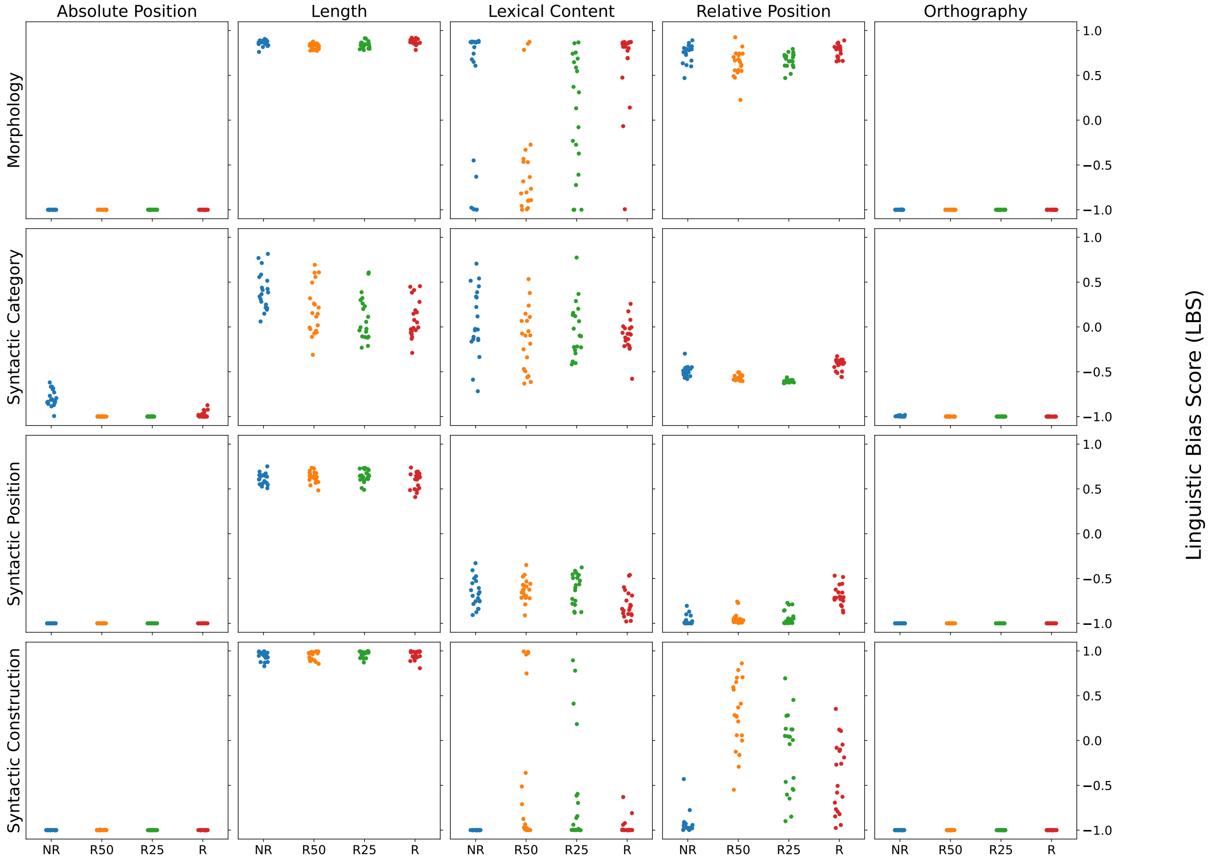


Figure 5: The dots in each sub-plot represent the LBS score of each run of each model. Each model has 20 different runs for each combination of surface and linguistic features. NR = Model pre-trained without retrieval, R50 = Model pre-trained with 50% noisy retrieval, R25 = Model pre-trained with 25% noisy retrieval, R = Model pre-trained with 0% noisy retrieval

E.6 LAMBADA

LAMBADA is a zero-shot language modeling tasks that focuses on resolving long-range dependencies in text (Paperno et al., 2016); we used its detokenized version from Radford et al. (2019). While it has been traditionally used for evaluating autoregressive language models, we adapt the task for masked language models. Note that this adaptation does not allow for a direct comparison with the autoregressive models. An illustrative sample from this dataset looks as follows:

Prompt: *"Give me a minute to change and I'll meet you at the docks." She'd forced those words through her teeth. "No need to change. We won't be that long." Shane gripped her arm and started leading her to the dock. "I can make it there on my own,* ***{answer}****."*

Gold answer: *Shane*

We insert the whole tokenized prompt to the evaluated language model and replace the missing answer by k mask tokens, where k is the length of the tokenized gold answer. Then we evaluate the exact-match accuracy of predicting filling in the correct continuation and also the mean perplexity.

Model	Accuracy	Perplexity
REFERENCE MODEL		
bert-base-cased	*44.77*	*26.95*
BASE		
− retrieval pretraining (patch)	**47.00**	**17.60**
− retrieval pretraining (no patch)	46.09	18.56
+ retrieval pretraining (50% noise, patch)	43.22	24.40
+ retrieval pretraining (25% noise, patch)	40.58	29.62
+ retrieval pretraining (0% noise, patch)	37.59	39.84
+ retrieval pretraining (0% noise, no patch)	22.63	141.62
SMALL		
− retrieval pretraining (patch)	35.11	44.81
− retrieval pretraining (no patch)	**35.84**	**41.25**
+ retrieval pretraining (0% noise, patch)	26.24	135.94
+ retrieval pretraining (0% noise, no patch)	0.43	37183.08
X-SMALL		
− retrieval pretraining (patch)	**25.42**	**133.44**
− retrieval pretraining (no patch)	25.33	137.73
+ retrieval pretraining (0% noise, patch)	19.33	329.90
+ retrieval pretraining (0% noise, no patch)	0.00	1.88×10^{11}

Table 8: Fine-grained LAMBADA results. The **bold** numbers represent the best model at each size, while the underline is the second best.

E.7 GLUE

To judge one of the facets of language understanding we use most of the GLUE benchmark (Wang et al., 2019). The benchmark is composed of the following tasks:

- **Corpus of Linguistic Acceptability** (CoLA; Warstadt et al., 2019) evaluated with the Matthews correlation coefficient (MCC; Matthews, 1975).
- **The Stanford Sentiment Treebank** (SST-2; Socher et al., 2013), evaluated with accuracy.
- **The Microsoft Research Paraphrase Corpus** (MRPC; Dolan and Brockett, 2005), evaluated with both F_1-score (originally also evaluated with accuracy).
- **The Quora Question Pairs** (QQP),[7] evaluated with F_1-score (originally evaluated with accuracy).
- **The Multi-Genre Natural Language Inference Corpus** (MNLI; Williams et al., 2018). Its development set consists of two parts: *matched*, sampled from the same data source as the training set, and *mismatched*, which is sampled from a different domain. Both parts are evaluated with accuracy.
- **Question-answering Natural Language Inference** (QNLI) constructed from the Stanford Question Answering Dataset (SQuAD; Rajpurkar et al., 2016), evaluated with accuracy.
- **The Recognizing Textual Entailment datasets** (RTE; Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), evaluated with accuracy.
- **The Semantic Textual Similarity Benchmark** (STS-B; Cer et al., 2017) is a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is human-annotated with a similarity score from 1 to 5; the task is to predict these scores. We evaluate using Pearson and Spearman correlation coefficients.
- **Winograd Schema Challenge** (WSC; Levesque et al., 2011) evaluated with accuracy.

[7] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

We omit the Winograd Schema Challenge due to the lack of training and test data leading to all our models underperforming compared to the majority label.

Table 9 shows the detailed results of each of the GLUE tasks. We see that independent of model size, the retrieval pre-trained models perform better on the CoLA dataset, although the difference between the models shrinks as the model size grows. In addition, we see inversions in the MNLI, RTE and STS-B tasks with the XS model performing better, the Small model on par and the Base model performing worse.

We did an extensive hyperparameter search for the retrieval pre-trained patched base and xs models as well as the regular pre-trained base and xs models. For the small version, we limited our learning rates to be in between those of the base and xs models. For the noisy versions, we combined the hyperparameters of the retrieval and regular pre-trained model and divided them by the amount of noise. In other words, the values of the learning rate for 25% noise are 25% of the way from the retrieval parameters going to the regular parameters, while keeping the batch size and warmup ratio the same as the retrieval version (although we made a mistake and did the opposite but to save compute, we have not re-run them correctly). For the 50% noise, we took the half-point values for all three hyperparameters. Finally, we used the hyperparameters of the base regular pre-trained models for BERT-BASE-CASED. The detailed list of the hyperparameters can be found in Table 10.

Model	**CoLA**	**SST-2**	**MRPC**	**QQP**	**MNLI**	**MNLI-mm**	**QNLI**	**RTE**	**STS-B**	**Average**
REFERENCE MODEL										
bert-base-cased	*57.4*$^{\pm 0.6}$	*91.3*$^{\pm 0.5}$	*89.2*$^{\pm 0.6}$	*87.2*$^{\pm 0.2}$	*82.5*$^{\pm 0.3}$	*82.9*$^{\pm 0.3}$	*89.2*$^{\pm 0.2}$	*63.9*$^{\pm 3.5}$	*88.9*$^{\pm 0.6}$/*88.5*$^{\pm 0.7}$	*82.1*$^{\pm 1.2}$
BASE										
− retrieval pretraining (patch)	$\underline{51.9}^{\pm 1.1}$	**91.8**$^{\pm 0.9}$	$\underline{90.5}^{\pm 0.4}$	**88.2**$^{\pm 0.1}$	$\underline{84.2}^{\pm 0.2}$	**84.4**$^{\pm 0.3}$	$\underline{91.4}^{\pm 0.3}$	62.1$^{\pm 3.8}$	**87.9**$^{\pm 0.3}$/**87.7**$^{\pm 0.3}$	$\underline{82.0}^{\pm 1.3}$
− retrieval pretraining (no patch)	$\underline{51.9}^{\pm 1.5}$	91.3$^{\pm 0.5}$	**90.6**$^{\pm 0.5}$	**88.2**$^{\pm 0.2}$	**84.4**$^{\pm 0.1}$	**84.4**$^{\pm 0.2}$	**91.5**$^{\pm 0.2}$	$\underline{64.4}^{\pm 3.9}$	$\underline{87.8}^{\pm 0.4}$/$\underline{87.6}^{\pm 0.4}$	**82.2**$^{\pm 1.4}$
+ retrieval pretraining (50% noise, patch)	51.7$^{\pm 1.5}$	91.2$^{\pm 0.9}$	90.3$^{\pm 0.9}$	$\underline{88.0}^{\pm 0.1}$	83.9$^{\pm 0.1}$	$\underline{83.9}^{\pm 0.1}$	91.3$^{\pm 0.1}$	**64.9**$^{\pm 3.5}$	87.7$^{\pm 0.3}$/87.5$^{\pm 0.3}$	$\underline{82.0}^{\pm 1.3}$
+ retrieval pretraining (25% noise, patch)	51.8$^{\pm 0.5}$	$\underline{91.4}^{\pm 0.2}$	**90.6**$^{\pm 0.6}$	87.9$^{\pm 0.1}$	83.9$^{\pm 0.3}$	83.8$^{\pm 0.2}$	91.1$^{\pm 0.1}$	63.5$^{\pm 1.4}$	87.7$^{\pm 0.4}$/87.4$^{\pm 0.4}$	81.9$^{\pm 0.6}$
+ retrieval pretraining (0% noise, patch)	51.4$^{\pm 1.8}$	91.3$^{\pm 0.8}$	90.1$^{\pm 1.2}$	87.8$^{\pm 0.2}$	83.3$^{\pm 0.1}$	83.4$^{\pm 0.2}$	90.2$^{\pm 0.3}$	61.1$^{\pm 3.6}$	86.8$^{\pm 0.3}$/86.6$^{\pm 0.3}$	81.2$^{\pm 1.4}$
+ retrieval pretraining (0% noise, no patch)	**53.1**$^{\pm 0.4}$	90.6$^{\pm 0.4}$	88.0$^{\pm 1.0}$	87.8$^{\pm 0.1}$	83.2$^{\pm 0.2}$	83.4$^{\pm 0.3}$	89.5$^{\pm 0.2}$	55.8$^{\pm 1.7}$	86.5$^{\pm 0.3}$/86.1$^{\pm 0.3}$	80.4$^{\pm 0.7}$
SMALL										
− retrieval pretraining (patch)	35.3$^{\pm 1.8}$	89.1$^{\pm 0.8}$	$\underline{88.3}^{\pm 1.2}$	$\underline{86.6}^{\pm 0.1}$	81.7$^{\pm 0.2}$	$\underline{82.0}^{\pm 0.3}$	$\underline{89.4}^{\pm 0.5}$	$\underline{53.4}^{\pm 3.3}$	84.2$^{\pm 0.5}$/83.8$^{\pm 0.5}$	77.4$^{\pm 1.3}$
− retrieval pretraining (no patch)	37.5$^{\pm 2.8}$	$\underline{89.8}^{\pm 0.5}$	**88.4**$^{\pm 0.7}$	**86.9**$^{\pm 0.1}$	**82.0**$^{\pm 0.1}$	**82.6**$^{\pm 0.1}$	**89.5**$^{\pm 0.3}$	53.3$^{\pm 2.3}$	**85.1**$^{\pm 0.5}$/**84.7**$^{\pm 0.5}$	$\underline{78.0}^{\pm 1.2}$
+ retrieval pretraining (0% noise, patch)	$\underline{40.4}^{\pm 2.1}$	**90.6**$^{\pm 0.5}$	$\underline{88.3}^{\pm 1.2}$	$\underline{86.6}^{\pm 0.1}$	$\underline{81.8}^{\pm 0.2}$	$\underline{82.0}^{\pm 0.2}$	89.0$^{\pm 0.3}$	**55.8**$^{\pm 1.4}$	**85.1**$^{\pm 0.4}$/**84.7**$^{\pm 0.4}$	**78.5**$^{\pm 0.9}$
+ retrieval pretraining (0% noise, no patch)	**40.9**$^{\pm 1.8}$	89.7$^{\pm 0.4}$	86.5$^{\pm 0.6}$	86.5$^{\pm 0.2}$	81.5$^{\pm 0.3}$	81.9$^{\pm 0.3}$	87.8$^{\pm 0.4}$	$\underline{53.4}^{\pm 2.0}$	$\underline{84.4}^{\pm 0.5}$/$\underline{84.1}^{\pm 0.4}$	77.7$^{\pm 0.9}$
X-SMALL										
− retrieval pretraining (patch)	$\underline{25.5}^{\pm 1.5}$	88.1$^{\pm 0.5}$	$\underline{88.3}^{\pm 0.7}$	84.6$^{\pm 0.2}$	78.3$^{\pm 0.2}$	79.3$^{\pm 0.2}$	86.4$^{\pm 0.2}$	51.1$^{\pm 4.7}$	82.4$^{\pm 0.5}$/82.0$^{\pm 0.5}$	74.6$^{\pm 1.6}$
− retrieval pretraining (no patch)	25.0$^{\pm 3.7}$	$\underline{88.6}^{\pm 0.4}$	**88.7**$^{\pm 0.9}$	**85.0**$^{\pm 0.1}$	78.8$^{\pm 0.3}$	79.7$^{\pm 0.1}$	**86.9**$^{\pm 0.4}$	$\underline{54.1}^{\pm 1.4}$	$\underline{82.8}^{\pm 0.2}$/$\underline{82.3}^{\pm 0.2}$	$\underline{75.2}^{\pm 1.3}$
+ retrieval pretraining (0% noise, patch)	**32.7**$^{\pm 2.4}$	$\underline{88.6}^{\pm 0.7}$	87.3$^{\pm 1.0}$	$\underline{84.9}^{\pm 0.1}$	**79.6**$^{\pm 0.3}$	$\underline{80.0}^{\pm 0.3}$	$\underline{86.8}^{\pm 0.2}$	**55.4**$^{\pm 2.2}$	82.5$^{\pm 0.7}$/$\underline{82.3}^{\pm 0.7}$	**76.0**$^{\pm 1.1}$
+ retrieval pretraining (0% noise, no patch)	25.4$^{\pm 2.2}$	**89.0**$^{\pm 0.6}$	85.0$^{\pm 1.0}$	84.7$^{\pm 0.2}$	$\underline{79.5}^{\pm 0.1}$	**80.2**$^{\pm 0.2}$	85.2$^{\pm 0.5}$	52.0$^{\pm 3.3}$	**82.9**$^{\pm 0.4}$/**82.7**$^{\pm 0.4}$	74.6$^{\pm 1.3}$

Table 9: Fine-grained GLUE results. The CoLA metric is MCC, the F1-score is used for MRPC and QQP, and the other tasks are evaluated with accuracy. Reported are the mean results and standard deviation from 5 seeded runs. The **bold** numbers represent the best model at each size, while the underline is the second best.

Hyperparameter	**CoLA**	**SST-2**	**MRPC**	**QQP**	**MNLI**	**QNLI**	**RTE**	**STS-B**
SHARED								
Epochs	10	10	10	4	4	10	10	10
Weight decay	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Learning Rate Scheduler	linear	linear	linear	linear	linear	linear	linear	linear
Attention Dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Classifier Dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Adam Epsilon	1e-6	1e-6	1e-6	1e-6	1e-6	1e-6	1e-6	1e-6
BASE − RETRIEVAL & REFERENCE MODEL								
Learning rate	2e-5	2e-5	5e-5	5e-5	5e-5	5e-5	1e-4	1.2e-4
Batch size	16	16	16	16	16	16	32	32
Warmup Ratio	0.1	0.06	0.1	0.06	0.1	0.06	0.06	0.1
BASE + RETRIEVAL(50% NOISE)								
Learning rate	3.5e-5	2e-5	7.5e-5	5e-5	5e-5	3.5e-5	1e-4	1.35e-4
Batch size	24	16	24	16	24	16	32	24
Warmup Ratio	0.08	0.06	0.1	0.08	0.1	0.08	0.06	0.1
BASE + RETRIEVAL (25% NOISE)								
Learning rate	2.75e-5	2e-5	6.25e-5	5e-5	5e-5	4.25e-5	1e-4	1.275e-4
Batch size	16	16	16	16	16	16	32	32
Warmup Ratio	0.1	0.06	0.1	0.06	0.1	0.06	0.06	0.1
BASE + RETRIEVAL								
Learning rate	5e-5	2e-5	1e-4	5e-5	5e-5	2e-5	1e-4	1.5e-4
Batch size	32	16	32	16	32	16	32	16
Warmup Ratio	0.06	0.06	0.1	0.1	0.1	0.1	0.06	0.1
SMALL − RETRIEVAL								
Learning rate	1.5e-4	2e-4	1e-4	1.5e-4	1e-4	5e-5	1e-4	1.8e-4
Batch size	32	32	8	32	32	16	8	8
Warmup Ratio	0.03	0.1	0.1	0.06	0.1	0.06	0.03	0.06
SMALL + RETRIEVAL								
Learning rate	1e-4	1e-4	1.25e-4	1e-4	1e-4	3e-5	1.25e-4	2e-4
Batch size	32	32	16	16	32	16	16	32
Warmup Ratio	0.03	0.06	0.06	0.06	0.06	0.06	0.06	0.12
XS − RETRIEVAL								
Learning rate	1.5e-4	2e-4	1e-4	1.5e-4	2e-4	5e-5	5e-5	2e-4
Batch size	16	16	32	16	32	16	8	8
Warmup Ratio	0.1	0.1	0.06	0.1	0.15	0.06	0.06	0.03
XS + RETRIEVAL								
Learning rate	1e-4	2.8e-4	1.5e-4	2.2e-4	1.8e-4	5e-5	1.5e-4	2e-4
Batch size	8	32	16	32	32	16	16	32
Warmup Ratio	0.12	0.1	0.06	0.06	0.1	0.1	0.06	0.06

Table 10: Fine-tuning hyperparameter details of GLUE, these are the optimal values found by the grid search described in Appendix E.7.

E.8 SQuAD

SQuAD is an extractive question answering dataset with 107,785 question-answer pairs. The task is to answer questions by providing the span of the correct answer string from a provided passage that is known to answer the question. We finetune all models over three epochs, using a learning rate of $5e-5$, a batch size of 16, and a weight decay of 0.01. Models are evaluated on the original development set, with no additional data used. We report the percentage of token-level exact matches (EM) and F1-score. The full set of results can be seen in Table 11.

We observe that retrieval impairs performance for all model sizes. For the base versions, the absolute performance decrease follow the amount of retrieved documents given to the model, showing that the closer one gets to a "perfect" set of retrieved documents, the worse the language model performs on the task of extractive QA. Furthermore, we observe that the addition of our patched linear layer has little effect on SQuAD for all model sizes, which we hypothesize is due to the size of the dataset; with over 100k examples, finetuning allows the model to fully "recover", making the patch obsolete.

Model	**Exact Match**	**F_1 score**
REFERENCE MODEL		
bert-base-cased	*80.6*$^{\pm0.2}$	*88.4*$^{\pm0.3}$
BASE		
− retrieval pretraining (patch)	**84.6**$^{\pm0.2}$	**91.3**$^{\pm0.1}$
− retrieval pretraining (no patch)	$\underline{84.4}^{\pm0.4}$	$\underline{91.2}^{\pm0.2}$
+ retrieval pretraining (50% noise, patch)	$83.9^{\pm0.1}$	$90.7^{\pm0.2}$
+ retrieval pretraining (25% noise, patch)	$83.3^{\pm0.5}$	$90.2^{\pm0.2}$
+ retrieval pretraining (0% noise, patch)	$82.8^{\pm0.1}$	$89.7^{\pm0.2}$
+ retrieval pretraining (0% noise, no patch)	$82.2^{\pm0.1}$	$89.7^{\pm0.2}$
SMALL		
− retrieval pretraining (patch)	$\underline{81.5}^{\pm0.2}$	**88.6**$^{\pm0.2}$
− retrieval pretraining (no patch)	**81.7**$^{\pm0.3}$	**88.6**$^{\pm0.2}$
+ retrieval pretraining (0% noise, patch)	$78.9^{\pm0.1}$	$\underline{86.3}^{\pm0.2}$
+ retrieval pretraining (0% noise, no patch)	$78.9^{\pm0.1}$	$86.2^{\pm0.2}$
X-SMALL		
− retrieval pretraining (patch)	$\underline{73.5}^{\pm0.2}$	**81.8**$^{\pm0.2}$
− retrieval pretraining (no patch)	**73.6**$^{\pm0.3}$	**81.8**$^{\pm0.2}$
+ retrieval pretraining (0% noise, patch)	$69.9^{\pm0.2}$	$\underline{78.7}^{\pm0.1}$
+ retrieval pretraining (0% noise, no patch)	$70.0^{\pm0.2}$	$\underline{78.7}^{\pm0.1}$

Table 11: Results on SQuAD 1.1. Results are reported as the mean and standard deviation over three random seeds. The **bold** numbers represent the best model at each size, while the $\underline{\text{underline}}$ is the second best.