Improving Natural Language Understanding with Computation-Efficient Retrieval Representation Fusion

Shangyu Wu[1]*, Ying Xiong[2]*, Yufei Cui[3]*†
Xue Liu[3], Buzhou Tang[2], Tei-Wei Kuo[45], Chun Jason Xue[1]
[1] City University of Hong Kong [2] Harbin Institute of Technology, Shenzhen
[3] MILA, McGill University [4] National Taiwan University
[5] Mohamed bin Zayed University of Artificial Intelligence

Abstract

Retrieval-based augmentations that aim to incorporate knowledge from an external database into language models have achieved great success in various knowledge-intensive (KI) tasks, such as question-answering and text generation. However, integrating retrievals in non-knowledge-intensive (NKI) tasks, such as text classification, is still challenging. Existing works focus on concatenating retrievals to inputs as context to form the prompt-based inputs. Unfortunately, such methods require language models to have the capability to handle long texts. Besides, inferring such concatenated data would also consume a significant amount of computational resources. To solve these challenges, we propose **ReFusion** in this paper, a computation-efficient **Re**trieval representation **Fusion** with neural architecture search. The main idea is to directly fuse the retrieval representations into the language models. Specifically, ReFusion first retrieves the representations of similar sentences and uses Neural Architecture Search (NAS) to seek the optimal fusion structures. Experimental results demonstrate our ReFusion can achieve superior and robust performance on various NKI tasks.

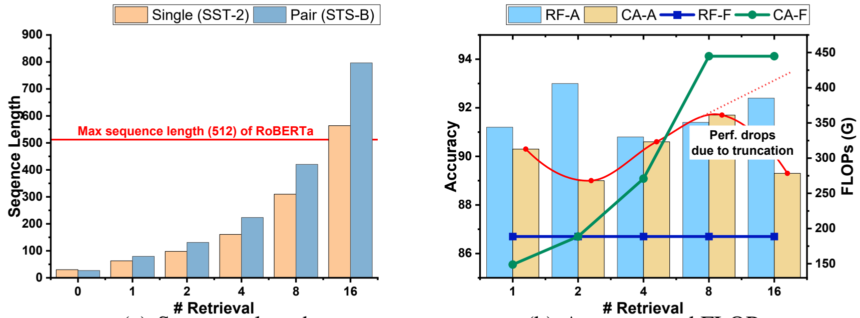
1 Introduction

Recent advances in language models (Khandelwal et al., 2020; Borgeaud et al., 2022; Guu et al., 2020; Lewis et al., 2020; Li et al., 2022) have demonstrated that retrieval-based augmentations can achieve remarkable performance on a variety of knowledge-intensive (KI) tasks. The basic idea of retrieval-based augmentations is to first leverage a dense vector indexing to retrieve the top-k related knowledge from an external database, then incorporate the retrieved knowledge into language models. For KI tasks such as question-answering and text generation, they have an inherent retrieval-based property (Chen et al., 2017; Karpukhin et al., 2020) as answers can be sourced or deduced from external knowledge databases.

However, retrieval-based augmentations in non-knowledge-intensive (NKI) tasks, such as text classification, are still challenging. Different from KI tasks, NKI tasks often require understanding and categorizing given sentences rather than generating new sentences (Wang et al., 2019). Previous works (Guo et al., 2023; Izacard & Grave, 2021) treat retrievals as the context of inputs and concatenate retrievals with inputs. However, their methods demand language models to have the capability of handling long sequence data. Figure 1(a) shows that concatenating more retrievals would significantly

*Authors contributed equally to this research
†Corresponding author

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) Sequence length (b) Accuracy and FLOPs

Figure 1: Impact of the number of retrievals on input sequence length and its effect on model's accuracy and FLOPs. CA refers to directly concatenating the retrievals with the input, while RF refers to directly adding the retrieval representations to the representation of the [CLS] token. CA-A and RF-A refer to the accuracy of context-augmentation and retrieval representation fusion. CA-F and RF-F refer to the FLOPs of context-augmentation and retrieval-addition.

increase the length of inputs, but the number of retrievals would be limited by the max sequence length of models. This limitation would result in a performance drop as shown in the red line in Figure 1(b). Besides, processing such long sequence inputs would also consume a substantial amount of computational resources as shown in the green line in Figure 1(b).

In this paper, we introduce **ReFusion**, a computation-efficient **Re**trieval representation **Fusion** framework with neural architecture search. Different from previous retrieval-based augmentations (Izacard & Grave, 2021; Guo et al., 2023), ReFusion directly fuses the representations of retrievals into models. ReFusion consists of three major modules, i.e., the retrieval module for retrieving neighbor representations, the fusion module for fusing the representations, and the search module for seeking the optimal combination of different fusion schemes. Experimental results on 15 NKI tasks show that ReFusion outperforms other comparisons and achieves superior and robust results. Codes are available at [3].

The main contributions of this paper are:

- We are the first to propose fusing the representations of retrievals directly into models to solve the performance and efficiency bottleneck of prompt-based techniques.
- Experimental results demonstrate that our ReFusion framework can significantly improve models' understanding capability, and achieve a superior and robust performance.

2 ReFusion: A Computation-Efficient Retrieval Representation Fusion with Neural Architecture Search

As shown in Figure 2(b), we propose a computation-efficient retrieval representation fusion framework. Our framework can be adapted to any transformer-based architecture (Vaswani et al., 2017), or any architecture that contains the attention module. The ReFusion contains three modules, i.e., the retrieval module for retrieving the representations of k similar sentences, the fusion module containing different fusion schemes, and the search module for seeking the optimal combination of different fusion schemes. Specifically, the retrieval module encodes the query texts and searches for the representations of top-k similar sentences among billions of data. The fusion module in this work involves different ranking schemes (e.g., a reranker-based scheme and an ordered-mask-based scheme Rippel et al. (2014); Cui et al. (2023, 2020, 2021); Mao et al. (2022)) to rerank the retrievals for different layers in LMs. Since it is difficult to tell which ranking scheme is better on each layer in LMs, the search module leverages neural architecture search (NAS) techniques to select the optimal ranking scheme or no ranking for each layer.

[3] https://anonymous.4open.science/r/ReFusion-173F

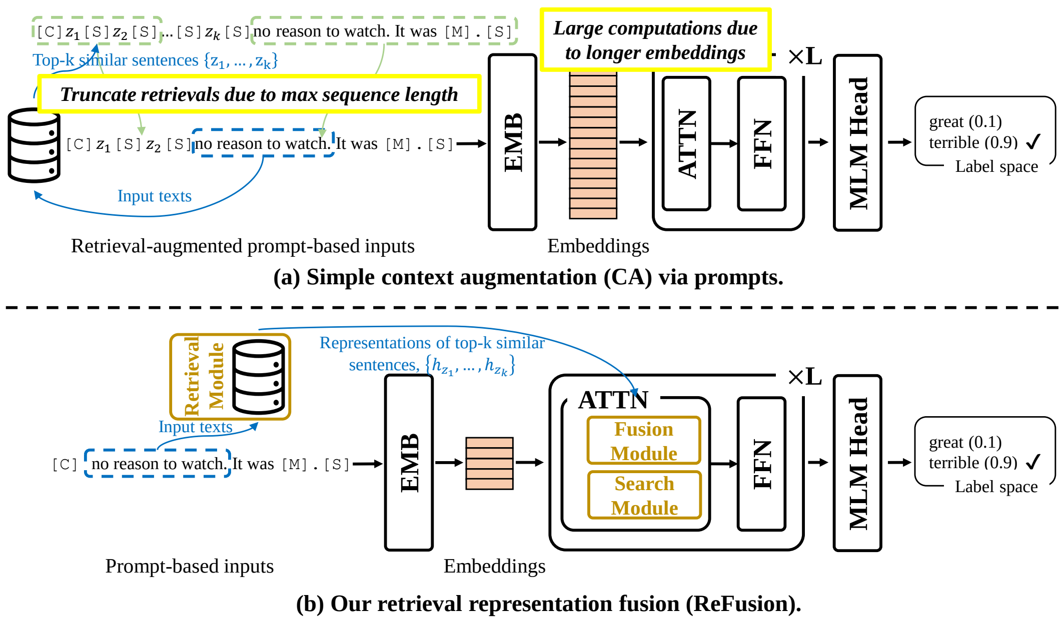


Figure 2: Retrieval-augmented prompt-based fine-tuning.

3 Experiment

3.1 Experimental Setting

Experimental Settings Our experimental setting mainly follows the settings in LM-BFF (Gao et al., 2021). We conduct comprehensive experiments across 15 NKI tasks, including 8 tasks from GLUE benchmark(Wang et al., 2019), SNLI, SST-5, MR, CR, MNLI, MNLI-mm, Subj and TREC. We measure the average performance of five different sampled D_{train} for each task with a fixed set of seed $S_{seed} = \{13, 21, 42, 87, 100\}$, which follows the LM-BFF's settings. Our models are based on RoBERTa-large for fair comparison with LM-BFF.

To validate the effectiveness of our method, we compared ReFusion with several other models: (1) LM-BFF: a prompt-based fine-tuning approach; (2) DART(Zhang et al., 2022): a differentiable prompt-based model, which can automatically search for the optimal prompt; (3) KPT(Hu et al., 2022): a prompt-based approach incorporating knowledge into the prompt verbalizer; and (4) CA-512: a retrieval-augmented prompt-based method concatenating retrievals with inputs.

3.2 Main Results

Table 1 presents the main experimental results of our ReFusion and comparisons on 15 NKI tasks. The results are shown in the form of means and variances, with the variance denoted by a subscript.

For tasks with single sentences (S-Task), ReFusion consistently demonstrates superior performance across almost all benchmarks. ReFusion achieves state-of-the-art performance on 5 tasks over 8 tasks. And ReFusion improves the average performance on the S-Task benchmark by about 2.1% than LM-BFF. Specifically, on the TREC task, ReFusion (90.3%) exhibits the maximum improvements over LM-BFF (84.8%).

For tasks consisting of pair sentences (P-Task), ReFusion continues to demonstrate strong performance. ReFusion also achieves the state-of-the-art on 5 tasks over 7 tasks. And ReFusion can improve the average performance on the P-Task benchmark by about 3.0% than LM-BFF. For instance, on the QNLI and SNLI benchmark, ReFusion (73% for QNLI, 80.6% for SNLI) significantly exceeds LM-BFF (64.5% for QNLI, 77.2% for SNLI).

The Avg-all represents the average performance of all 15 NKI tasks. For overall average performance, ReFusion achieves a score of 74.3%, marginally surpassing LM-BFF's 71.8%. This further highlights

Table 1: Our main results with RoBERTa-large.

Methods	SST-2	SST-5	MR	CR	MPQA	SUBJ	TREC	CoLA	Avg-S
LM-BFF	$92.7_{0.9}$	$47.4_{2.5}$	$87.0_{1.2}$	$90.3_{1.0}$	$84.7_{2.2}$	$91.2_{1.1}$	$84.8_{5.1}$	$9.3_{7.3}$	73.4
DART	$\mathbf{93.5_{0.5}}$	-	$\mathbf{88.2_{1.0}}$	$\mathbf{91.8_{0.5}}$	-	$90.7_{1.4}$	$87.1_{3.8}$	-	-
KPT	$90.3_{1.6}$	-	$86.8_{1.8}$	$88.8_{3.7}$	-	-	-	-	-
CA-512	$91.3_{1.4}$	$46.7_{1.1}$	$85.1_{1.4}$	$88.3_{1.7}$	$76.9_{2.8}$	$88.0_{1.9}$	$82.2_{4.4}$	$7.4_{3.3}$	70.7
ReFusion	$93.4_{0.6}$	$\mathbf{49.8_{1.4}}$	$87.9_{1.1}$	$91.7_{0.3}$	$\mathbf{86.7_{1.1}}$	$\mathbf{92.5_{0.8}}$	$\mathbf{90.3_{3.7}}$	$\mathbf{11.4_{4.1}}$	**75.5**
Methods	**MNLI**	**MNLI-m**	**SNLI**	**QNLI**	**RTE**	**MRPC**	**QQP**	**Avg-P**	**Avg-all**
LM-BFF	$68.3_{2.3}$	$70.5_{1.9}$	$77.2_{3.7}$	$64.5_{4.2}$	$69.1_{3.6}$	$74.5_{5.3}$	$65.5_{5.3}$	69.9	71.8
DART	$67.5_{2.6}$	-	$75.8_{1.6}$	$66.7_{3.7}$	-	$\mathbf{78.3_{4.5}}$	$67.8_{3.2}$	-	-
KPT	$61.4_{2.1}$	-	-	$61.5_{2.8}$	-	-	$\mathbf{71.6_{2.7}}$	-	-
CA-512	$66.2_{1.0}$	$67.8_{1.3}$	$71.6_{2.2}$	$66.9_{3.2}$	$66.6_{3.1}$	$73.5_{6.9}$	$64.0_{1.9}$	68.1	69.5
ReFusion	$\mathbf{69.3_{1.5}}$	$\mathbf{70.9_{1.5}}$	$\mathbf{80.6_{1.4}}$	$\mathbf{73.0_{1.1}}$	$\mathbf{70.9_{2.3}}$	$77.0_{3.6}$	$68.9_{3.3}$	**72.9**	**74.3**

The results of LM-BFF, DART refer to their original paper. The results of KPT refer to Chen et al. (2022). The numbers are the average results. The subscript numbers are the standard deviation results.

ReFusion's consistent and superior performance. Besides, ReFusion surpasses other models like DART, CA-512 and KPT, delivering superior or comparable results. Notably, the standard deviation of ReFusion is considerably smaller than that of other models, indicating that ReFusion produces stable results and offers superior robustness.

3.3 Ablation Study

We conduct ablation experiments on six representative tasks to show the contributions of each module to the overall performance. On all tasks, ReFusion tends to produce better results than those just applying the retrieval fusion module. The results of methods using NAS demonstrate that NAS can significantly boost performance. Specifically, compared to the baseline, two ranking schemes can bring different but significant improvements. This reveals that it is necessary to combine different ranking schemes on different tasks. After using NAS, the performance of each ranking scheme is also significantly improved. This suggests these two ranking schemes are not always suitable for every layer in LMs, thus we need to disable the fusion module at some layers. Finally, our ReFusion integrating all effective candidate fusion modules using NAS achieves the best performance on three tasks. We can infer that the combination of all candidate modules harnesses their strengths.

Table 2: Ablation studies on different modules.

Methods	MPQA	SUBJ	TREC	SNLI	QNLI	RTE
Roberta-Large	$83.6_{2.5}$	$90.3_{2.8}$	$83.8_{5.2}$	$73.5_{5.2}$	$65.0_{3.0}$	$64.1_{2.0}$
Reranker	$84.2_{2.2}$	$91.3_{1.3}$	$85.0_{4.2}$	$74.3_{4.6}$	$68.8_{1.4}$	$65.6_{3.1}$
Ordered Mask	$83.3_{1.9}$	$90.8_{1.4}$	$83.0_{5.8}$	$74.9_{4.0}$	$68.3_{1.4}$	$65.8_{3.1}$
NAS with Reranker	$86.9_{1.3}$	$92.4_{1.3}$	$\mathbf{90.8_{2.5}}$	$80.3_{1.9}$	$\mathbf{73.5_{1.8}}$	$69.2_{2.4}$
NAS with Ordered Mask	$\mathbf{87.0_{1.5}}$	$92.4_{0.7}$	$90.7_{3.0}$	$80.3_{1.3}$	$73.0_{1.0}$	$70.4_{2.5}$
ReFusion	$86.7_{1.1}$	$\mathbf{92.5_{0.8}}$	$90.3_{3.7}$	$\mathbf{80.6_{1.4}}$	$73.0_{1.1}$	$\mathbf{70.9_{2.3}}$

The numbers are the average results. The subscript numbers are the standard deviation results.

4 Conclusion

In this paper, we aim to solve the bottleneck of prompt-based techniques by directly fusing retrieval representations into models. We propose a computation-efficient retrieval representation fusion framework with neural architecture search, ReFusion. ReFusion uses NAS to fuse retrievals refined by different ranking schemes on each layer in LMs. Experimental results demonstrate our fusion framework outperforms baselines and is robust on various tasks.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2206–2240. PMLR, 2022.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1870–1879, 2017.

Xiang Chen, Lei Li, Ningyu Zhang, Xiaozhuan Liang, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. RETROPROMPT-Decoupling Knowledge from Memorization: Retrieval-augmented Prompt Learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23908–23922. Curran Associates, Inc., 2022.

Yufei Cui, Ziquan Liu, Wuguannan Yao, Qiao Li, Antoni B. Chan, Tei-Wei Kuo, and Chun Jason Xue. Fully nested neural network for adaptive compression and quantization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 2080–2087, 2020.

Yufei Cui, Ziquan Liu, Qiao Li, Antoni B. Chan, and Chun Jason Xue. Bayesian nested neural networks for uncertainty calibration and adaptive compression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 2392–2401, 2021.

Yufei Cui, Yu Mao, Ziquan Liu, Qiao Li, Antoni B. Chan, Xue (Steve) Liu, Tei-Wei Kuo, and Chun Jason Xue. Variational nested dropout. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(8): 10519–10534, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 3816–3830. Association for Computational Linguistics, 2021.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3887–3896. PMLR, 2020.

Zhicheng Guo, Sijie Cheng, Yile Wang, Peng Li, and Yang Liu. Prompt-guided retrieval augmentation for non-knowledge-intensive tasks. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 10896–10912. Association for Computational Linguistics, 2023.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909, 2020.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pp. 2225–2240. Association for Computational Linguistics, 2022.

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

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 6769–6781, 2020.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Zonglin Li, Ruiqi Guo, and Sanjiv Kumar. Decoupled context processing for context augmented language modeling. In *NeurIPS*, 2022.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019b.

Yu Mao, Yufei Cui, Tei-Wei Kuo, and Chun Jason Xue. Accelerating general-purpose lossless compression via simple and scalable parameterization. In João Magalhães, Alberto Del Bimbo, Shin'ichi Satoh, Nicu Sebe, Xavier Alameda-Pineda, Qin Jin, Vincent Oria, and Laura Toni (eds.), *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pp. 3205–3213. ACM, 2022.

Oren Rippel, Michael A. Gelbart, and Ryan P. Adams. Learning ordered representations with nested dropout. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pp. 1746–1754. JMLR.org, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

A Templates on All Tasks

Table 3 provides an overview of the manual templates and selected label words used for each dataset in our experiments. These templates and label words were created following LM-BFF (Gao et al., 2021).

Table 3: Templates and label words that we used in our experiments.

Task	Prompts	Label word
SST-2	`[CLS]` x It was `[MASK]`. `[SEP]`	"0":"terrible", "1":"great"
SST-5	`[CLS]` x It was `[MASK]`. `[SEP]`	"0":"terrible","1": "bad", "2": "okay","3": "good","4": "great"
MR	`[CLS]` x It was `[MASK]`. `[SEP]`	"0":"terrible", "1":"great"
CR	`[CLS]` x It was `[MASK]`. `[SEP]`	"0":"terrible", "1":"great"
MPQA	`[CLS]` x It was `[MASK]`. `[SEP]`	"0":"terrible", "1":"great"
SUBJ	`[CLS]` x This is `[MASK]`. `[SEP]`	"0":"subjective", "1":"objective"
TREC	`[CLS]` `[MASK]` x `[SEP]`	"0":"Description","1":"Entity","2":"Expression", "3":"Human","4":"Location","5":"Number"
CoLA	`[CLS]` x It was `[MASK]`. `[SEP]`	"0":"incorrect", "1":"correct"
MNLI	`[CLS]` x_1 ? `[MASK]`, x_2 `[SEP]`	"contradiction": "No","entailment":"Yes", "neutral": "Maybe"
MNLI-m	`[CLS]` x_1 ? `[MASK]`, x_2 `[SEP]`	"contradiction": "No","entailment":"Yes", "neutral": "Maybe"
SNLI	`[CLS]` x_1 ? `[MASK]`, x_2 `[SEP]`	"contradiction": "No","entailment":"Yes", "neutral": "Maybe"
QNLI	`[CLS]` x_1 ? `[MASK]`, x_2 `[SEP]`	"not entailment":"No ","entailment":"Yes"
RTE	`[CLS]` x_1 ? `[MASK]`, x_2 `[SEP]`	"not entailment":"No ","entailment":"Yes"
MRPC	`[CLS]` x_1 `[MASK]`, x_2 `[SEP]`	"0":"No", "1":"Yes"
QQP	`[CLS]` x_1 `[MASK]`, x_2 `[SEP]`	"0":"No", "1":"Yes"

B Results on Full Training Set

We conduct experiments on several tasks under the prompt-based setting with the full training set. As shown in Table 4, across all datasets, ReFusion generally demonstrates either comparable or superior performance compared to LM-BFF. The average performance across all tasks in ReFusion surpasses that of LM-BFF by 1.4%. This suggests that ReFusion's performance superiority is consistent and not dependent on the size of the dataset. This implies that ReFusion is robust and can generalize well across varying amounts of data.

Table 4: Full training set results compared with LM-BFF.

Methods	SST-2	SST-5	MR	CR	MPQA	SUBJ	TREC	CoLA	RTE
LM-BFF	95.0	58.7	90.8	89.4	87.8	97.0	97.4	62.6	80.9
ReFusion	**95.6**	**61.0**	**92.3**	**91.4**	**84.4**	**97.1**	**97.6**	**62.8**	**85.2**

C Technique Details

C.1 The Online Retrieval Module

In the retrieval module, there is a query encoder for encoding query texts and a task-agnostic retriever built offline over billions of dense vectors. The retriever consists of an efficient indexing like FAISS (Johnson et al., 2019) or ScaNN (Guo et al., 2020), and a compressed key-value store database that contains all texts and embeddings. The retrieving process in our framework is online performed, which means that for every forward, the query encoder first passes the representation h_x of the input

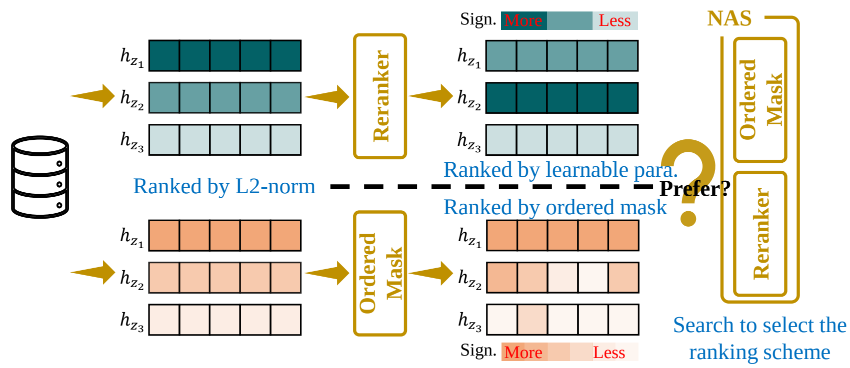


Figure 3: Two different ranking schemes used in the fusion module.

text x to the retriever, then the retriever returns the representations $H_Z = \{h_{z_1}, \ldots, h_{z_k}\}$ of top-k similar sentences $Z = \{z_1, \ldots, z_k\}$ to the fusion module. For efficient retrieving, especially for the training, the retrieval module maintains an in-memory cache for the input text x and corresponding representations H_Z of similar sentences.

C.2 The Retrieval Fusion Module

The retrieval fusion module can be wrapped with any modules in the language models (LMs). It takes the representations of top-k similar sentences and the hidden representations of existing modules as inputs, and outputs the fused representations. Specifically, we introduce two effective ranking schemes as shown in Figure 3.

C.2.1 Reranking the Retrievals

In the retrieval module, the retrievals are ranked by a task-agnostic similarity metric, e.g., L2 norm. Directly adding the representations to the hidden representations would not improve LMs' performance. That is because 1) The retrievals are not optimally ranked for the existing module in LMs, which may introduce noise or irrelevant information; 2) The models should pay different attention to those retrievals in case of overemphasizing less relevant information. Therefore, we aim to propose a learnable reranker to learn the ranking distribution tailored to each module in LMs. As shown in the top of Figure 3, the significance of retrievals is re-assigned after reranking.

Specifically, the reranker is a 1D learnable vector of k dimensions, i.e., $R = \{r_1, \ldots, r_k\}$. It is first normalized and then multiplied by the retrievals. Finally, the averaged representation of all reranked retrievals is added to the sentence representation, e.g., `[CLS]` token in BERT-like models (Liu et al., 2019b; Devlin et al., 2019). The formal steps are as follows,

$$r_i = \frac{\exp(r_i)}{\sum_j \exp(r_j)} \tag{1}$$

$$h_{y_{\texttt{[CLS]}}} = h_{x_{\texttt{[CLS]}}} + \frac{1}{k}\sum r_i \cdot h_{z_i} \tag{2}$$

where $h_{x_{\texttt{[CLS]}}}$, $h_{y_{\texttt{[CLS]}}}$ are the sentence representations of inputs and outputs.

C.2.2 Ordered Mask Over Retrieval Representations

Rippel et al. (Rippel et al., 2014) proposed a nested dropout that directly drops the representation units from the sampled index I, thus yielding an inherent importance ranking of the representation dimensions. This nested dropout can be implemented by a mask with leading I ones then zeros. Based on the nested dropout, recent works (Cui et al., 2023, 2020, 2021; Mao et al., 2022) proposed the ordered mask that modeled the dropping process with a chain of Bernoulli variables and made it differentiable using the re-parameterization trick.

As shown in the bottom of Figure 3, we apply the ordered mask over k retrievals on each representation dimension. This means that different from the reranker, the ordered mask trusts the ranking produced by the retriever and refines the ranking with training data. Specifically, let $h_{z_1}, \ldots, h_{z_k}$ be the top-k D-dimensional retrieval representations. For each dimension of retrieval representation (e.g., the dimension d), the ordered mask is modeled by a chain of Bernoulli variables $V = \{v_1^d, \ldots, v_k^d\}$, where $v_i^d \sim$ **Bernoulli**(π_i) indicates whether drop the d-th representation unit of the i-th retrieval. Following the property of nested dropout, the variable v_i^d is conditioned on v_{i-1}^d, thus we can obtain the marginal distribution $p(\mathbf{v_i^d})$ of v_i^d.

After that, the ordered mask uses the re-parameterization trick, e.g., choosing the Gumbel Softmax distribution Jang et al. (2017) as the tractable variational distribution $q(\mathbf{v_i^d})$. With Gumbel Softmax distribution, if $\mathbf{c}^d \sim$ **Gumbel**(β, τ), then $v_i^d = 1 - \texttt{cumsum}_i(\mathbf{c}^d)$, where $\mathbf{c}^d$ is a sample choice of the dropped index over k retrievals on the dimension d, and $\texttt{cumsum}_i(\mathbf{c}^d) = \sum_{j=0}^{i-1} c_j^d$. In the Gumbel Softmax distribution, β is a learnable parameter in the differentiable function $v_i^d = g(\epsilon_i; \beta)$ and τ is the temperature variable that controls the smoothness of the step at the dropped index.

Finally, we obtain the different ordered mask $V^1, \ldots, V^D$ over representation dimensions. We use it to mask the retrievals in a fine-grained way. Then, the masked retrievals would be fused into the sentence representations in the same way as Reranker. The formal steps are as follows,

$$\mathbf{c}^d \sim \textbf{Gumbel}(\beta, \tau) \tag{3}$$

$$v_i^d = 1 - \texttt{cumsum}_i(c^d) \tag{4}$$

$$\hat{h}_{z_i}^d = v_i^d \cdot h_{z_i}^d \tag{5}$$

$$h_{y_{\texttt{[CLS]}}} = h_{x_{\texttt{[CLS]}}} + \frac{1}{k} \sum \hat{h}_{z_i} \tag{6}$$

where $\hat{h}_{z_i}^d$ is the d-th masked representation unit of i-th retrieval.

C.3 The Architecture Search Module

As shown in Figure 3, it is difficult to tell which ranking scheme is better on each layer in LMs. Therefore, we propose an architecture search module, aiming to leverage neural architecture search (NAS) techniques to search to select the optimal ranking scheme.

C.3.1 Search Space

In this work, we do not search for a totally new neural network architecture like previous NAS works (Liu et al., 2019a) do. Instead, we keep the main structure of transformer-based models unchanged and only replace several modules with our search modules.

A search module consists of multiple fusion modules with different ranking schemes and the original module. For example, taking the linear module in LMs as an example, we replace the linear module with our linear search module, which includes three modules, the fusion module with reranker-based scheme, the fusion module with ordered-mask-based scheme, and the original linear module.

Although the number of candidate modules in the search module is small, the whole search space is quite large. Given a transformer-based language model with N hidden layers, assume that we only replace the linear module for the key and value in every attention module, we have at least $3 \times 3 = 9$ candidate modules and thus at least 9^N different retrieval-augmented transformer-based language models. Taking the RoBERTa-large as an example, which has 24 layers, the search space can be septillion-level large.

C.3.2 Searching Details

We follow the same searching strategies used in DARTS (Liu et al., 2019a). Specifically, let $\alpha = \{\alpha_1, \ldots, \alpha_l\}$ be the architectural weights, where l is the number of candidate modules in each search module. To make the search space continuous, we also relax the categorical choice of a particular candidate module to a softmax over all possible candidate modules within the search module,

$$\hat{o}(h) = \sum_i \frac{\exp(\alpha_i)}{\sum_j \exp(\alpha_j)} o_i(h) \tag{7}$$

where $o_i(h)$ represents the output of the i-th candidate module $o_i(\cdot)$ taking the hidden states h as input, $\hat{o}(\cdot)$ indicates the output of the search module.

The goal of architecture searching is to jointly optimize the architectural weights α and the weights ω of all modules with LMs. We update the weights ω based on the training loss, and the architectural weights based on the validation loss. The updates of these two types of weights are done alternatively. After training, we only choose the candidate module with the largest architectural weights for the inference.