ERATE: Efficient Retrieval Augmented Text Embeddings

Vatsal Raina* **Nora Kassner** **Kashyap Popat**
Patrick Lewis **Nicola Cancedda** **Louis Martin**
Meta AI
vr311@cam.ac.uk louismartin@meta.com

Abstract

Embedding representations of text are useful for downstream natural language processing tasks. Several universal sentence representation methods have been proposed with a particular focus on self-supervised pre-training approaches to leverage the vast quantities of unlabelled data. However, there are two challenges for generating rich embedding representations for a new document. 1) The latest rich embedding generators are based on very large costly transformer-based architectures. 2) The rich embedding representation of a new document is limited to only the information provided without access to any explicit contextual and temporal information that could potentially further enrich the representation. We propose efficient retrieval-augmented text embeddings (ERATE) that tackles the first issue and offers a method to tackle the second issue. To the best of our knowledge, we are the first to incorporate retrieval to general purpose embeddings as a new paradigm, which we apply to the semantic similarity tasks of SentEval. Despite not reaching state-of-the-art performance, ERATE offers key insights that encourages future work into investigating the potential of retrieval-based embeddings.

1 Introduction

State-of-the-art sentence embedding models (Raffel et al., 2020; Neelakantan et al., 2022) have competed against one another to approach human-like performance in several NLP tasks. Despite the gains observed in performance of sentence embeddings on public benchmarks such as SentEval (Conneau and Kiela, 2018a), the progress has come at a large computational expense. For example, the largest model amongst the Sentence-T5 series consists of up to billions of parameters while GPT-3 based sentence embedding model released by Neelakantan et al. (2022) has 175 billion parameters with marginal gains observed in performance when compared against older, smaller models. Models of these sizes are compute intensive and very difficult to host and use for most downstream use cases.

We propose a new paradigm that aims to maintain the benefits of high-complexity rich embedding models at reduced computational requirements. Our novel paradigm investigates whether retrieval can be used to bypass the compute intensive embedding model in a similar manner to the application of retrieval for generation (Lewis et al., 2020; Cai et al., 2022) tasks for real world large scale use cases with latency and compute constraints. We propose to use a lightweight retrieval model combined with rich pre-computed representations, in order to approximate the richer representations of a large embedding model.

We find retrieval-based embeddings struggle against standard text embedding models but their performance can be improved by aggregating neighbours from different light embedding representations and increasing the size of the datastore of precomputed embeddings.

To our knowledge, this is the first attempt to use retrieval approaches for developing general purpose sentence embeddings. Our main contributions can be summarised as follows:

- Introduction of a novel paradigm for generating sentence embeddings by exploiting retrieval-based approaches.
- Releasing efficient retrieval augmented text embeddings (ERATE) baseline systems with an exploration of methods that work well and don't work as well to assess the scope of retrieval to recover the performance of rich embedding models with low compute.

[*] Work done during internship.

The Fourth Workshop on Insights from Negative Results in NLP, pages 11–18
May 5, 2023 ©2023 Association for Computational Linguistics

We hope other researchers will engage in this novel setup to develop more efficient sentence embeddings that will allow high-performing representations to be accessible to a broader community.

Our work focuses on developing lightweight embeddings that out-compete existing lightweight embeddings but we believe ERATE can be used for a wider range of applications. Specifically, input documents often lack the full contextual information or temporal relevance to generate the necessary high-quality text embedding. ERATE offers the opportunity for the embedding of a given document to encapsulate information from other similar documents to increase the information content whilst also being more up-to-date with more recent documents added to a datastore.

2 Related Work

Reimers and Gurevych (2019) introduced Sentence-BERT as an improvement to the sentence representations from BERT (Devlin et al., 2019) by explicitly training Siamese BERT-networks using pairs of similar/dissimilar sentences. Yan et al. (2021) released ConSERT to learn sentence representations in an unsupervised manner by applying various forms of augmentations to a sentence to create its pair for contrastive learning. In a similar vain, SimCSE (Gao et al., 2021) relied on unsupervised contrastive learning by using dropout masks as the augmentation technique. DiffCSE (Chuang et al., 2022) further incorporated masked language modelling as an augmentation technique. Ni et al. (2022), released a family of sentence-T5 models that finetuned the T5 (Raffel et al., 2020) architecture in a supervised manner with pairs of naturally occurring similar sentences. Most recently, Neelakantan et al. (2022) developed a model finetuned using GPT-3 (Brown et al., 2020).

Several works have looked at approaches to make less expensive sentence embedding representations. For example, embedding recycling (Saad-Falcon et al., 2022) for language models is proposed as a reduced compute approach for downstream tasks. This involves caching activations from intermediate layers in large pre-trained models such that when similar inputs are seen during inference time, the cached output can be used in order to skip a part of the model structure. Embedding recycling has been demonstrated to out-compete distilled models, such as DistilBERT (Sanh et al., 2019). In contrast, we investigate whether fixed embedding representations can be generated more efficiently using retrieval without any additional training, relying only on pre-computed embeddings.

Other works have investigated efficient methods for retrieval from a large set of documents such as ColBERT (Khattab and Zaharia, 2020) and PLAID (Santhanam et al., 2022) interaction models that use offline encoding of documents. Rather than making the retrieval step more efficient, our work focuses on using retrieval as a tool for enhancing the development of general purpose embeddings.

Text generation and language modelling has seen several works involving performance boost with retrieval. Khandelwal et al. (2019) investigates extending a pre-trained language model by including the k-nearest neighbours, which Kassner and Schütze (2020) applies to question-answering. Similarly, Lewis et al. (2020) introduced retrieval-augmented generation (RAG) models where a pre-trained retriever and a pre-trained sequence-to-sequence model are fine-tuned end-to-end. Borgeaud et al. (2022) released RETRO as a successor of REALM (Guu et al., 2020) where an autoregressive language work is retrieval-enhanced by making the training documents explicitly available at inference time. Finally, Izacard et al. (2022) present ATLAS for retrieval-enhanced language modelling where the sequence-to-sequence model takes the retrieved documents and the query to generate the output text for knowledge-intensive tasks. We probe whether retrieval-incorporated approaches can bring similar benefits for the development of fixed embedding representations, not end-to-end sequence-to-sequence models.

3 Retrieval for text embeddings

This section explains how efficient retrieval augmented text embeddings are developed. The main idea is that a query document only needs to be embedded using a light embedder and by outlining the nearest neighbours in the light space, the corresponding pre-computed embeddings can be combined to generate the rich query embedding.

Let $\hat{d}$ denote a new document, for which we want to determine the rich embedding representation, $\hat{\mathbf{x}}$. Let $f_{\text{light}}(\cdot)$ and $f_{\text{rich}}(\cdot)$ be embedding generators that map a given document to the light and rich embedding spaces respectively:

$$\mathbf{h} = f_{\text{light}}(d) \qquad \mathbf{x} = f_{\text{rich}}(d) \tag{1}$$

Note, we assume that the operation $f_{\text{rich}}(d)$ is pro-

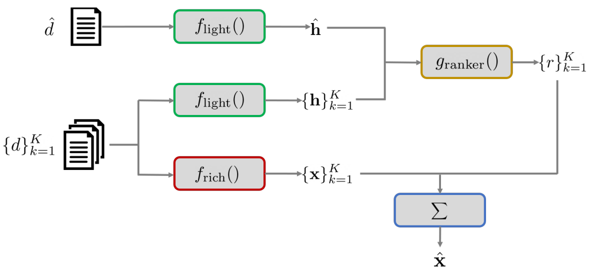


Figure 1: Schematic for ERATE embedding generation.

hibitively compute intensive while $\hat{\mathbf{h}} = f_{\text{light}}(\hat{d})$ is feasible. Instead, there exists a set of documents $\{d\}_{k=1}^{K}$ for which the rich embeddings, $\{\mathbf{x}\}_{k=1}^{D}$, have been pre-computed. Let $g_{\text{ranker}}(\cdot)$ denote a retrieval system that ranks all embeddings (with pairwise cosine distance) in a set based upon a query embedding. Hence, the ranks are:

$$\{r\}_{k=1}^{K} = g_{\text{ranker}}(\hat{\mathbf{h}}; \{\mathbf{h}\}_{k=1}^{K}) \tag{2}$$

The final rich embedding can then be calculated as a combination of the rich embedding representations of the top R documents:

$$\hat{\mathbf{x}} = \frac{1}{R}\sum_{k=1}^{K} \mathbf{1}_{r_k \leq R} \cdot \mathbf{x}_k \tag{3}$$

The process is depicted in the pipeline of Figure 1. Alternative approaches can be considered for the combination process of Equation 3 [1].

3.1 Dropout masks

The proposed set-up for ERATE relies on identifying neighbours to the query document in the light space. However, the set of neighbours identified in the light space are correlated with the light embedding model that may not necessarily align with the desired neighbours in the rich space. Consequently, it is useful to create a neighbour set curated from multiple light embedding models which reduces the bias to a single light embedder (see Figure 2).

Dropout (Srivastava et al., 2014) is a common regularisation technique that has been extended to create diverse outputs at inference time such as Monte Carlo dropout (Gal and Ghahramani, 2016). Similarly, randomly *dropping* out embedding dimensions can be used to create a diverse set of light embedders that can expect to have different, potentially complementary, neighbour sets. Therefore dropout masks are applied to the light embeddings prior to performing retrieval in the ERATE process to create enchanced neighbour sets.

4 Experiments

4.1 Setup

SentEval (Conneau and Kiela, 2018b) is a popular benchmark dataset for assessing the quality of sentence embeddings, consisting of semantic text similarity (STS) tasks STS-12 to STS-16 and STS-B, SICK-R. These tasks evaluate how well the cosine distances of embeddings from pairs of sentences correlate with human annotated similarity scores using Spearman's rank correlation coefficient[2].

For ERATE to work effectively, a large datastore of documents/sentences must exist for which the sentence embeddings must be pre-calculated using both a light embedder and a rich embedder. We select the average GloVe word embeddings[3] (Pennington et al., 2014) as the light embedder as the model involves a simple lookup for each word in the sentence to determine its word embedding and hence low compute. State-of-the-art performance on the STS tasks of SentEval is achieved by Sentence-T5-xxl[4] (Ni et al., 2022). Hence, we adopt this Sentence-T5 model as our rich embedder. Additionally, we consider an *Oracle* ERATE model to breakdown the retrieval and combination stages of ERATE embeddings. Oracle embeddings are calculated by retrieving the closest neighbours in the rich space instead of the light space.

	Wiki	SNLI	MNLI	CC
# sentences	1M	629K	519K	100M
avg. words	$19_{\pm 12}$	$8_{\pm 4}$	$12_{\pm 9}$	$25_{\pm 19}$

Table 1: Statistics for unique datastore sentences.

The datastore of sentences with pre-computed embeddings is constructed from combining the 1 million Wikipedia (Wiki) sentences that acted as the unsupervised training data for SimCSE (Gao et al., 2021) and DiffCSE (Chuang et al., 2022) with the unique sentences of the *premise* and *hypothesis* from the SNLI (Bowman et al., 2015) and

[1] Empirical experiments indicated that weighing the importance of a retrieved embedding by its inverse distance to the query in the light space did not improve performance and hence the simplest approach of a linear average was adopted.

[2] Consistent with previous works, the 'all' setting that aggregates the subsets in a given STS task is used from https://github.com/facebookresearch/SentEval.

[3] Available at: https://huggingface.co/sentence-transformers/average_word_embeddings_glove.840B.300d

[4] Available at: https://huggingface.co/sentence-transformers/sentence-t5-xxl

MNLI (Williams et al., 2018) datasets. An additional 100 million sentences sampled from common crawl (CC)[5] are included in an expanded datastore to investigate the impact of increasing the datastore size. Table 1 details the statistics for each of these subsets. Sentences from STS on average have $13_{\pm 10}$ words, which is of a similar length to the sentences that are being used for the datastore as well as in terms of the diversity of topics.

4.2 Results

For a 512 token sentence the vanilla ERATE model (with a datastore size of 1 million) requires 3×10^9 floating point operations (FLOPs) while the Sentence-T5 model requires 8.7×10^{12} FLOPs.

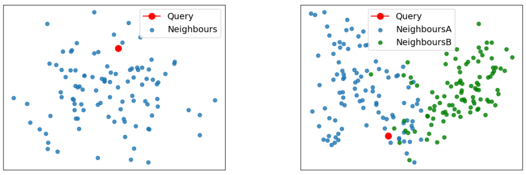
Table 2 presents the performance of the baseline ERATE system against the existing state-of-the-art performance from Sentence-T5. Using the compute intensive rich embeddings directly achieves an average correlation coefficient of 84.8% across all the STS tasks while the light embedding model achieves a performance of 62.8% at a fraction of the compute. In contrast, the ERATE embeddings (100 closest neighbours are selected in the retrieval step), which have a similar compute to the light embedder, only achieve 55.3%. This low performance is underwhelming as let alone being close to state-of-the-art, it is not able to compete against the light embedding model.

	Avg.	sts12	sts13	sts14	sts15	sts16	stsB	sickR
Rich	84.8	78.9	88.9	84.9	89.3	84.7	86.7	80.4
Light	62.8	57.5	71.0	60.7	70.8	63.8	60.9	54.8
Oracle	72.3	66.8	76.9	70.9	73.6	73.7	75.2	69.1
ERATE	55.3	57.2	59.7	47.3	59.9	54.5	53.8	54.7
+drop.	57.4	60.8	62.0	52.8	59.8	54.4	56.5	55.4
+expand	57.9	55.4	60.0	52.9	64.1	60.0	58.8	53.8

Table 2: Performance with Sentence-T5 (Rich), GloVe (Light), oracle neighbours and vanilla ERATE with dropout and an expanded datastore.

The significant boost in performance to 72.3% from the Oracle suggests that the combination process by averaging is somewhat successful and the loss in performance comes from a mismatch in the surrounding neighbours for the light vs rich space. Further work would benefit from investigating alignment between the light and rich spaces.

Figure 2a further depicts an example PCA plot (using the two most dominant dimensions). Here, the rich embedding of an example query sentence is compared to the rich embeddings of the closest neighbours identified from the light space. On observation [6], the query lies on the periphery of the neighbours, which leads to the the centroid of the neighbours being afar from the desired query's position. We confirm the anisotropy hypothesis as the ratio of the distance between the query to the centroid and the averaged neighbour distance to the centroid (averaged across all test examples) is $1.1_{\pm 0.4}$ while the equivalent ratio using the Oracle neighbours is $0.5_{\pm 0.2}$ - about twice as close.



(a) Query vs neighbours. (b) Neighbours with dropout.

Figure 2: PCA on rich embeddings showing the query is closer to the centroid with multiple neighbour sets.

Consequently, as discussed in Section 3.1, an expanded neighbour set is considered by applying different dropout masks on 50% of the dimensions. Visually, Figure 2b suggests that the neighbour set from each dropout mask is somewhat different and hence the centroid of all the neighbours is more likely to approach the query's rich embedding. The hypothesis is supported by Table 2 where the performance increases to 57.4% by using 10 dropout masks simultaneously.

The performance can expect to be higher if the neighbours of the query are from a dense region as the combination of the embeddings will have less error. Therefore, Table 2 details the performance when using an expanded datastore size consisting of an additional 100 million sentences from Common Crawl (see Table 1). The baseline ERATE system performance is boosted by 2.5%.

5 Ablations

This section presents three ablations: (1) using an alternative light embedder; (2) an attempt to align the light and rich embedding spaces; (3) distillation of a rich embedder onto a light embedder.

Table 2 presents the results of ERATE where the average GloVe embeddings are used for the light embedder and the Sentence-T5-xxl model is used as the rich embedder. Here, an alternative light embedder is considered: the embedding associated

[5] https://commoncrawl.org/

[6] Observed on several examples.

with the [CLS] token of the DistilBERT (Sanh et al., 2019) model[7]. From Table 3, the ERATE approach successfully out-competes the DistilBERT light embedder by an encouraging 3.7% but it is still worse performing than the ERATE approach with the average GloVe embedder from Table 2.

	Avg.	sts12	sts13	sts14	sts15	sts16	stsB	sickR
Rich	84.8	78.9	88.9	84.9	89.3	84.7	86.7	80.4
Light*	39.6	32.1	38.0	31.3	44.1	52.8	31.0	47.7
ERATE*	45.0	37.6	34.3	51.1	47.0	43.1	50.1	44.0

Table 3: Performance with Sentence-T5 (Rich), DistilBERT (Light*), oracle neighbours and vanilla ERATE*.

ERATE relies on combining the rich embeddings of the neighbours identified from a light embedding space. Table 2 showed that the Oracle neighbours from the rich space substantially out-compete ERATE. Hence, it is expected that if the neighbour sets between the light and rich spaces have greater agreement, there will be improved performance for ERATE. A projection system is trained from the average GloVe embedding space to the ST5-xxl embedding space for better alignment.

Spaces	$P@1$	$P@10$	$P@100$
GloVe vs ST5	13.31	13.92	15.64
Projected[GloVe] vs ST5	12.51	13.10	14.33

Table 4: Impact of aligning light and rich spaces with a projection layer using Precision@K for $K \in \{1, 10, 100\}$.

The projection model consists of an input layer followed by a ReLU followed by a single hidden layer that predicts an embedding in the target embedding space with a cosine embedding loss. The vanilla datastore embeddings are used as the training data with 10% of the data cut-out for validation. Table 4 assesses the improved alignment by applying the projection layer. The averaged Precision@K is used as an assessment metric that measures the fraction of the closest K neighbours that match in each space for a given query. Despite that the model is trained to project the light space onto the rich space, there is degradation in the alignment of neighbours, possibly because the ordering of surrounding neighbours is not maintained in the training regime that impacts the retrieved neighbours.

A distillation inspired approach is considered where a light embedding model aims to mimic the embeddings of the rich Sentence-T5 model as an alternative strategy to ERATE. DistilBERT is selected as the light model[8]. For every datastore embedding, the light model is finetuned (all parameters) to predict the output embedding from the rich model. The distilled model achieves an average score on the STS tasks of 45.6% which is lower than the light model from Table 3. The lower performance may occur due to no emphasis on maintaining semantic similarity explicitly.

6 Conclusions

Retrieval-based embeddings are proposed as ERATE that bypass inference through an expensive embedding generation model but hope to leverage its richness. However, the current set-up for ERATE achieves subpar performance on text similarity tasks with some gains observed from combining neighbours of a unique dropout mask approach and extending the datastore size of pre-computed light and rich embeddings for retrieval. We highlight multiple areas of future work.

Future work should investigate ERATE-based approaches in a hybrid setting: ERATE embeddings are used for sentences where they are likely to work effectively (neighbours are in a dense space allowing accurate approximations) while the default expensive embedder can be used when ERATE is unlikely to be effective. ERATE can be increasingly effective when only partial information is available in a query for which an embedding is desired as combining the embeddings of neighbouring documents can enrich the information content. However, sentence-level embeddings offer little opportunity to explore the gains by additional information and hence future work should investigate the scope of ERATE at the document-level; MTEB (Muennighoff et al., 2022) potentially offers suitable tasks. We should also investigate alternative approaches for aligning the light and rich spaces and better combining neighbours' embeddings e.g. self-attention.

References

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.

[7] Available at: https://huggingface.co/distilbert-base-uncased

[8] The GloVe model is not used as there is no availability to finetune the model.

Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, pages 2206–2240. PMLR.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Deng Cai, Yan Wang, Lemao Liu, and Shuming Shi. 2022. Recent advances in retrieval-augmented text generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3417–3419.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. *arXiv preprint arXiv:2204.10298*.

Alexis Conneau and Douwe Kiela. 2018a. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexis Conneau and Douwe Kiela. 2018b. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

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.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-wei Chang. 2020. Realm: Retrieval-augmented language model pre. *Training*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Nora Kassner and Hinrich Schütze. 2020. Bert-knn: Adding a knn search component to pretrained language models for better qa. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3424–3430.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

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.

Jon Saad-Falcon, Amanpreet Singh, Luca Soldaini, Mike D'Arcy, Arman Cohan, and Doug Downey. 2022. Embedding recycling for language models. *arXiv preprint arXiv:2207.04993*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022. Plaid: an efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1747–1756.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

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.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075.

.

Appendix A Limitations

The experiments for ERATE are currently limited to the semantic text similarity tasks of SentEval. More comprehensive experiments should investigate the applicability of ERATE against benchmark text embedding representations for a wide range of downstream NLP tasks.

Appendix B Computational resources

All experiments were conducted using NVIDIA A100 graphical processing units.

Appendix C Reproducibility

The experiments conducted in this work has only relied on publicly available data and publicly available models. There was no additional training of models. Additional hyperparameters for ERATE embeddings (e.g. the size of the datastore, the number of neighbours, the dropout rate) is detailed in the relevant sections of the main paper.

Appendix D Licenses

This section details the license agreements of the scientific artifacts used in this work. The Stanford Natural Language Inference (SNLI) Corpus is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. For MNLI, the majority of the corpus is released under the OANC's license, which allows all content to be freely used, modified, and shared under permissive terms. The data in the FICTION section falls under several permissive licenses; Seven Swords is available under a Creative Commons Share-Alike 3.0 Unported License, and with the explicit permission of the author, Living History and Password Incorrect are available under Creative Commons Attribution 3.0 Unported Licenses; the remaining works of fiction are in the public domain in the United States (but may be licensed differently elsewhere). SentEval is released under the BSD License. Common Crawl is released under the MIT License.

Appendix E Additional experiments

In the main paper, ERATE relies on combining the rich embedding representations of the neighbours that have been identified using the light embedding representations. The number of neighbours was set to 100. In this section, the impact on the downstream STS tasks is investigated when a different number of neighbours are considered instead. Table Appendix E.1 details the performance when using a different number of neighbours from the datastore. The best averaged results are observed empirically when 100 neighbours are used from the datastore.

#neigh.	Avg.	sts12	sts13	sts14	sts15	sts16	stsB	sickR
1	40.9	30.2	40.7	35.1	50.8	43.7	39.8	46.0
10	52.4	51.6	52.9	43.5	61.9	49.1	53.8	54.0
100	**55.3**	57.2	59.7	47.3	59.9	54.5	53.8	54.7
1000	54.7	54.2	58.8	48.6	61.3	54.0	52.3	53.8
10,000	52.4	51.2	57.3	46.9	59.1	50.9	49.1	52.4

Table Appendix E.1: Varying the number of neighbours for ERATE.